2mif

20029-H221-RO-00

(NASA-CR-134238) CONIC PROGRAM USER'S MANUAL (TRW Systems) ~~48~~ 42 p HC $5.50 CSCL 09B

N74-20843

Unclas
G3/08 36068

2d
JND

TRW NOTE NO. 73-FMT-927

CR 134238

CONIC PROGRAM USER'S MANUAL

October 1973

Prepared for

MISSION PLANNING AND ANALYSIS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
JOHNSON SPACE CENTER
HOUSTON, TEXAS

NAS 9-12330

A. G. Flores
A. G. Flores
Vehicle Dynamics Section

T. E. Lane
T. E. Lane, Task Manager
JSC/TRW Task 427
Vehicle Dynamics Section

J. J. Vaccaro
J. J. Vaccaro, Manager
Systems Dynamics Department

L. W. Neel
L. W. Neel
Assistant Project Manager
Systems Performance Analysis
Mission Trajectory Control Project

TRW
SYSTEMS

ABSTRACT

This report is the User's Manual for the current CONIC computer program, written for GDB under Task 427. This document revises and supersedes TRW Note 70-FMT-835, CONIC Program User's Manual, May 27, 1970. The program has been written in FORTRAN for use on the TRW Time Share system. The CONIC program is used to calculate the relative motion of two bodies in any conic-section orbit (circular, elliptical, parabolic, or hyperbolic).

This manual describes the basic equations used for trajectory propagation and defines the input and output parameters of the program. Also included are sample runs with corresponding figures and a detailed flow chart of the program.

TABLE OF CONTENTS

1.0 INTRODUCTION

The CONIC Program is designed for the calculation of the relative trajectories of two vehicles in orbit, where the orbits may be circular, elliptical, parabolic, or hyperbolic. The inertial trajectories of the two vehicles are calculated independently as conic sections, with no limitation on the type of orbit.

The only inputs required by the program are (1) specification of the gravitational constant of the attracting body, (2) the initial positions and velocities of the primary and secondary vehicles, and (3) a list of the maneuvers to be performed by the secondary body. The maneuvers are specified by phase, where each phase consists of an impulsive maneuver of specific delta velocity and direction, followed by a nonpropulsive coast of specified duration. A maximum of ten such maneuver phases may be defined for a single run.

The output of the program is the relative position and velocity of the secondary vehicle relative to the primary vehicle, printed at equal intervals during the coasting flight. The number of intervals per phase is controlled by program input.

The program is ideally suited to the calculation of relative motion in a highly elliptical orbit such as a semi-synchronous earth orbit, or in a hyperbolic orbit such as a lunar flyby or an approach to earth entry. It is also very accurate in computing the relative motion for complex maneuvers in circular orbit, such as a rendezvous-rescue sequence. It is limited to impulsive maneuvers, and should not be used for long-term low-level thrusts such as continuous venting or atmospheric drag.

2.0 BASIC EQUATIONS

The basic equation by which the trajectories of the bodies are calculated is the Kepler equation for nonpropulsive motion in a conic section (ellipse, parabola, or hyperbola), expressed in terms of universal variables. Use of this equation allows direct solution of the problem of predicting the position and velocity of a body after a given time interval "t", given its initial position and velocity. Kepler's equation may be written as:

$$\sqrt{\mu} \quad t = \frac{\vec{r_o} \cdot \vec{v_o}}{\sqrt{\mu}} \; x^2 \; C(\alpha_o x^2) + (1 - r_o \alpha_o) x^3 \; S(\alpha_o x^2) + r_o x \qquad (1)$$

where the functions $S(\xi)$ and $C(\xi)$ are defined as:

$$S(\xi) = \frac{1}{3!} - \frac{\xi}{5!} + \frac{\xi^2}{7!} -\ldots = \begin{cases} \dfrac{\sqrt{\xi} - \sin\sqrt{\xi}}{\left(\sqrt{\xi}\right)^3} \; \ldots\ldots\ldots \; \xi > 0 \\ \\ \dfrac{\sinh\sqrt{-\xi} - \sqrt{-\xi}}{\left(\sqrt{-\xi}\right)^3} \; \ldots \; \xi < 0 \end{cases} \qquad (2)$$

$$C(\xi) = \frac{1}{2!} - \frac{\xi}{4!} + \frac{\xi^2}{6!} -\ldots = \begin{cases} \dfrac{1 - \cos\sqrt{\xi}}{\xi} \; \ldots\ldots\ldots \; \xi > 0 \\ \\ \dfrac{\cosh\sqrt{-\xi} - 1}{-\xi} \; \ldots\ldots\ldots \; \xi < 0 \end{cases} \qquad (3)$$

and α_o is defined by the equation:

$$\alpha_o = \frac{2}{r_o} - \frac{{v_o}^2}{\mu} \qquad (4)$$

and

- μ is the gravitational constant
- t is the time interval of propagation
- $\vec{r}_o$ is the initial position vector of the body
- $\vec{v}_o$ is the initial velocity vector of the body
- r_o is the magnitude of r_o
- v_o is the magnitude of v_o
- x is the normalized anomaly

An examination of Kepler's equation shows that it is an explicit expression for the known time of propagation "t" in terms of the unknown variable "x". To evaluate x for a given value of t requires an inversion of the function $t = t(x)$ given by Equation 1. This is done in a straightforward manner by employing Newton's method, as follows: given an estimate $x_n = x_n(t_n)$ of the required value of $x = x(t)$, an improved estimate x_{n+1} is obtained from:

$$x_{n+1} = x_n - \frac{\sqrt{\mu}\ t_n - \sqrt{\mu}\ t}{\left(\sqrt{\mu}\ \frac{dt}{dx}\right)_{x = x_n}} \tag{5}$$

where the derivative is given by:

$$\sqrt{\mu}\ \frac{dt}{dx} = \frac{\vec{r}_o \cdot \vec{v}_o}{\mu}\ [x - \alpha_o x^3 S(\alpha_o x^2)] + (1 - r_o\alpha_o)x^2 C(\alpha_o x^2) + r_o \tag{6}$$

The process is repeated until a sufficiently accurate value of x is obtained. Once x has been determined, the position and velocity vectors of the body at time t are given by:

$$\vec{r}(t) = \left[1 - \frac{x^2}{r_o}\ C(\alpha_o x^2)\right]\vec{r}_o + \left[t - \frac{x^3}{\sqrt{\mu}}\ S(\alpha_o x^2)\right]\vec{v}_o \tag{7}$$

$$\vec{v}(t) = \frac{\sqrt{\mu}}{rr_o}\left[\alpha_o x^3 S(\alpha_o x^2) - x\right]\vec{r}_o + \left[1 - \frac{x^2}{r}\ C(\alpha_o x^2)\right]\vec{v}_o \tag{8}$$

where r is the magnitude of $\vec{r}(t)$.

The foregoing method for propagating the state vector of a body is applied independently to the primary and secondary bodies to obtain their inertial state vectors at the required times. The relative motion parameters are then obtained by subtracting the state vector of the primary body from that of the secondary body, and expressing the resultant relative vector in a local-vertical-local horizontal system based on the primary vehicle. The process is repeated for each printout in a given coast phase. At the beginning of a new phase, the delta velocity vector of the impulsive maneuver is converted to inertial coordinates and added to the inertial velocity of the secondary vehicle. The resulting state vectors of the primary and secondary bodies are then propagated for the duration of the phase.

3.0 INPUT PARAMETERS

A total of seven input options have been included in CONIC to allow for maximum flexibility in providing input to the program. These options are as follows:

Input option 1 (INOPT1) allows a choice of coordinate systems. If INOPT1 is set equal to 1, the relative position and velocity of the secondary body is expressed in a local-vertical-local-horizontal Cartesian coordinate system centered on the primary body. If INOPT1 is set equal to 2, the relative position and velocity of the primary and secondary bodies are converted to state vectors in an inertial coordinate system.

The second input option (INOPT) allows a choice of methods for specifying the pitch direction of maneuvers executed by the secondary body. If INOPT is set equal to 1, the pitch maneuvers are measured upward from the direction of the posigrade local horizontal. This method of specifying the pitch of a maneuver is used both for elliptical orbits and for hyperbolic orbits. If INOPT is set equal to 2, the pitch maneuvers are measured upward from the direction of the inertial velocity vector of the secondary body. This method of specifying the pitch of a maneuver is most useful for hyperbolic orbits such as lunar flyby or earth entry orbits.

Input option 2 (INOPT2) allows a choice of output units. If INOPT2 is set equal to 1, the relative position of the secondary body will be typed out in nautical miles. If INOPT2 is set equal to 2, it will be typed out in feet.

Input option 3 (INOPT3) is used to define the gravitational constant (EMU) and radius (RO) of the central gravitational body. If INOPT3 is set at 1, an earth orbit is assumed, and if INOPT3 is set at 2, a lunar orbit is assumed. Direct input of EMU and RO is not required, because the necessary values are included in the program.

Input option 4 (INOPT4) is used to choose the method for initializing the position and velocity of the primary (or reference) body. Setting INOPT4 equal to 1 allows for direct input of the inertial position (XOP) and velocity (VOP) vectors of the primary body in feet and feet per second. If INOPT4 is set equal to 2, the primary body is assumed to be in a circular orbit, and the only required input parameter is the altitude of the circular orbit (H) in nautical miles. The state vector for the primary body is then calculated automatically according to the formulas:

```
XOP(1) = 0.0                      VOP(1) = SQRT(EMU/XOP(2))
XOP(2) = RO + H * 6076.1155       VOP(2) = 0.0
XOP(3) = 0.0                      VOP(3) = 0.0
```

Regardless of how the initial state vector for the primary body is input, it is used to calculate the type of orbit in which the primary body will move. If the orbit is hyperbolic or parabolic, a message is typed to identify the orbit type. If the orbit is elliptical, the period is computed and typed. The orbital period is needed when the coast times between maneuvers are known only in orbit fractions.

Input option 5 (INOPT5) allows a choice of methods for initializing the position (XOS) and velocity (VOS) of the secondary body. If INOPT5 is set equal to 1, the initial position and velocity of the secondary body are set equal to that of the primary body. If INOPT5 is set equal to 2, the initial position and velocity of the secondary body may be specified relative to the primary body in a local-vertical-local-horizontal coordinate system based on the primary body. This coordinate system is defined with the "+X" axis in the local horizontal posigrade direction, the "+Y" axis in the local vertical upward direction, and the "Z" axis completing a right-handed system. If INOPT5 is set equal to 3, the inertial state vector of the secondary body may be input directly. This option is available only if the primary body was also initialized by direct specification of the inertial state vector (INOPT4 = 1).

Input option 6 (INOPT6) is encountered at the end of each case, and provides a choice of three ways to begin the next case. When the program is executed from statement 1, the first required input is INOPT3, which allows the gravitational body to be specified as either the earth or the moon. The next required inputs are the initial state vectors of the two bodies. The last required inputs are the maneuvers to be executed by the secondary body. If INOPT6 is set equal to 1, execution of the next case begins with the input of new initial state vectors and continues with the input of a new table of maneuvers. If INOPT6 is set equal to 2, execution of the next case begins with the input of a new table of maneuvers, leaving the initial state vectors unchanged. If INOPT6 is set equal to 3, the program branches to STOP. This option would be used when no more cases are to be executed, or when the next case must begin execution at statement 1 in order to change the value of INOPT3, which specifies the gravitational body.

In addition to the inputs provided for by the various input options, the maneuvers to be executed by the primary or secondary body must also be supplied. Six parameters must be input for each phase, where each phase consists of an impulsive maneuver followed by coasting flight. The impulsive maneuver is executed either by the primary or secondary body (B). The impulsive maneuver is defined by the pitch (P) and yaw (Y) directions of the maneuver and its magnitude (DV). The pitch (P) is input in degrees, measured upward from either the local horizontal posigrade or the inertial velocity vector, depending on whether INOPT was set equal to 1 or 2, respectively. The yaw (Y) is also input in degrees, measured positively to the right of the orbital plane (facing in the direction of the inertial velocity). The magnitude of the impulsive maneuver (DV) is input in feet per second. The coasting flight following the impulsive maneuver is specified by the total coast time (CT), input in seconds, and the number of points (NP) to be plotted per phase. The total input for each phase consists of the primary or secondary body (B), pitch (P), yaw (Y), and delta velocity (DV) of the

impulsive maneuver, and the coast time (CT) and number of points to be plotted (NP) for the subsequent coasting flight. The inputs for pitch (P), yaw (Y), delta velocity (DV), coast time (CT), and number of points (NP) must be in decimal form. Any number of phases may be specified (NPHASE), up to a maximum of ten phases. The input for the specific number of maneuver phases must be in I3 (INTEGER) format, i.e., one maneuver phase is inputted as 001.

4.0 OUTPUT PARAMETERS

The output parameters for CONIC consist of the relative position and velocity of the secondary body expressed in a local-vertical-local-horizontal Cartesian coordinate system centered on the primary body. The definitions of the output parameters are as follows:

T......... Time duration measured from the beginning of the first phase (seconds)

RPDR...... Relative position downrange of the secondary body, positive in the direction of the inertial velocity of the primary body (nautical miles or feet)

RPRA...... Relative position radial of the secondary body, positive upward (nautical miles or feet)

RPCR...... Relative position crossrange of the secondary body, positive to the right of the orbital plane when facing in the direction of the inertial velocity (nautical miles or feet)

RVDR...... Relative velocity downrange of the secondary body (feet per second)

RVRA...... Relative velocity radially up of the secondary body (feet per second)

RVCR...... Relative velocity crossrange of the secondary body (feet per second)

5.0 SAMPLE COMPUTER RUNS

The following listings and figures show the input and output of several computer runs. The output data is plotted on Figures 1 and 2 for the corresponding inputs. The input parameters required by the program and supplied by the user are underlined; all of the remaining print-out is typed automatically by the program. The input options for Figure 1 are the same as those for Figure 2. The first four input options are set equal to 1. This sets the options so that a local-vertical-local-horizontal coordinate system is assumed, pitch maneuvers are measured upward from the local horizontal posigrade, the relative position is output in nautical miles, and so that an earth orbit is assumed. The primary body is initialized by specification of its circular orbit altitude. The secondary body is initialized with its relative position and velocity equal to that of the primary body.

Immediately after initialization of the primary body, its type of orbit is calculated, and the orbit type is identified if it is hyperbolic or parabolic. The orbital period is also calculated and printed out. The period is needed because coast times between maneuvers must be input in seconds. This allows the specification of coast times equal to a period or even to a fraction of a period. All of these sample runs are executed for one maneuver phase.

The maneuver table allows a detailed specification of the maneuvers to be performed either by the primary or secondary body. The maneuvers are specified by phases, where each phase consists of an impulsive maneuver, followed by a coasting flight during which the relative position and velocity of the specific body are printed out. The maneuvers in both Figures 1 and 2 were performed on a coasting flight of 10,000.0 seconds with printout at 10 equal intervals during the coast.

The output blocks show the result of the input parameters. The first line of output shows the relative position and velocity of the secondary vehicle before the first maneuver. Since the secondary vehicle was initialized with its position and velocity equal to that of the primary vehicle,

its relative position and velocity are zero. The second line of output (the first line under Phase Number 1) shows the relative position and velocity of the secondary vehicle immediately after the phase 1 maneuver. The remaining lines of output under phase 1 give the relative position and velocity of the secondary body for ten equal increments during a one-orbit coast. The listings of maneuvers I, IIa, and IIIa in Figure 1 were randomly selected as sample print-outs.

Three sample posigrade separation maneuvers at two different delta velocities are plotted in Figure 1. All three maneuvers were executed in-plane (0 degree yaw) and at 3.3 feet per second and 7.0 feet per second, respectively. In maneuvers I and Ia, a 0 degree pitch is specified. A pitch of 45 degrees upward is specified in maneuvers II and IIa. A pitch of 90 degrees (radially upward) is specified in maneuvers III and IIIa. The relative motions of the last two maneuvers return to the original position at the end of one orbit resulting in football maneuvers. Figure 1 shows only the side views of the relative motion of the several maneuvers.

Two sample posigrade separation maneuvers at different delta velocities are plotted in Figure 2. Both maneuvers were executed with 30 degrees pitch upward and 30 degrees yaw out of plane. Maneuver I was performed at 3.3 feet per second and Maneuver II at 7.0 feet per second. Figure 2 shows both the side and top views of the relative motion of the two maneuvers.

Note that the relative position is still zero, but the relative velocity is now upward and out of plane, as specified by the input. The remaining lines of output under phase 1 give the relative position and velocity of the secondary body for ten equal increments during a one-orbit coast. The in-plane motion is a football maneuver, returning to the original position at the end of one orbit.

In the second run, all of the input options were set equal 2, so that a lunar orbit is assumed, the primary body is initialized by specification of its circular orbit altitude, the secondary body is initialized by specifying its position and velocity relative to the primary body, the pitch of maneuvers is measured upward from the inertial velocity vector of the secondary body, and so that relative position is output in feet. The maneuver specified is 10 feet per second radially up (football maneuver), with output at ten points equally spaced over one orbit.

INPUT OPTIONS for FIGURES 1 and 2

```
INOPT1(1:LVLH COORD, 2:INERTIAL COORD)
? 1
INOPT(1:PITCH MEASURED UP FROM LOCAL HORIZONTAL POSIGRADE)
      (2:PITCH MEASURED UP FROM INERTIAL VELOCITY VECTOR)
? 1
INOP12(1:OUTPUT IN N. MILES,2:OUTPUT IN FEET)
? 1
INOP13(1:EARTH ORBIT,2:LUNAR ORBIT)
? 1
INOP14(INITIALIZATION OPTION FOR PRIMARY BODY)
1:INPUT STATE VECTOR,2:INPUT ALTITUDE OF CIRCULAR ORBIT
? 2
H(CIRCULAR ORBIT ALTITUDE OF PRIMARY BODY,N.M.)
? 235.
PERIOD OF PRIMARY BODY =   5596.99 SECONDS
INOPTS(INITIALIZATION OPTION FOR SECONDARY BODY)
1:SAME AS PRIMARY,2:INPUT RELATIVE VECTOR
? 1
NPHASE(NUMBER OF MANEUVER PHASES, MAX OF 10)(FORMAT I3)
? 001
```

FIGURE 1 - MANEUVER I

```
MANEUVER TABLE
BODY(1 OR 2),PITCH(DEG),YAW(DEG),DELTA V(FPS),COAST(SEC),NO.OF POINTS)
B,P,Y,DV,CT,NP FOR PHASE 1
? 2
? 0.
? 0.
? 3.3
? 10000.
? 10.
```

T (SEC)	RPDR (NM)	RPRA (NM)	RPCR (NM)	RVDR (FPS)	RVRA (FPS)	RVCR (FPS)
0	0.0	0.0	0.0	0.000	0.000	0.000
PHASE NUMBER	1					
0	0.0	0.0	0.0	3.300	0.000	0.000
1000	.1	.5	0.0	-4.180	5.948	0.000
2000	-1.7	1.6	0.0	-18.139	5.157	0.000
3000	-5.3	1.9	0.0	-22.771	-1.469	0.000
4000	-8.4	1.2	0.0	-12.837	-6.432	0.000
5000	-9.4	.2	0.0	.424	-4.112	0.000
6000	-8.9	.1	0.0	1.987	2.870	0.000
7000	-9.5	1.0	0.0	-9.921	6.601	0.000
8000	-12.2	1.8	0.0	-21.804	2.859	0.000
9000	-15.9	1.7	0.0	-20.215	-4.115	0.000
10000	-18.2	.8	0.0	-6.951	-6.430	0.000

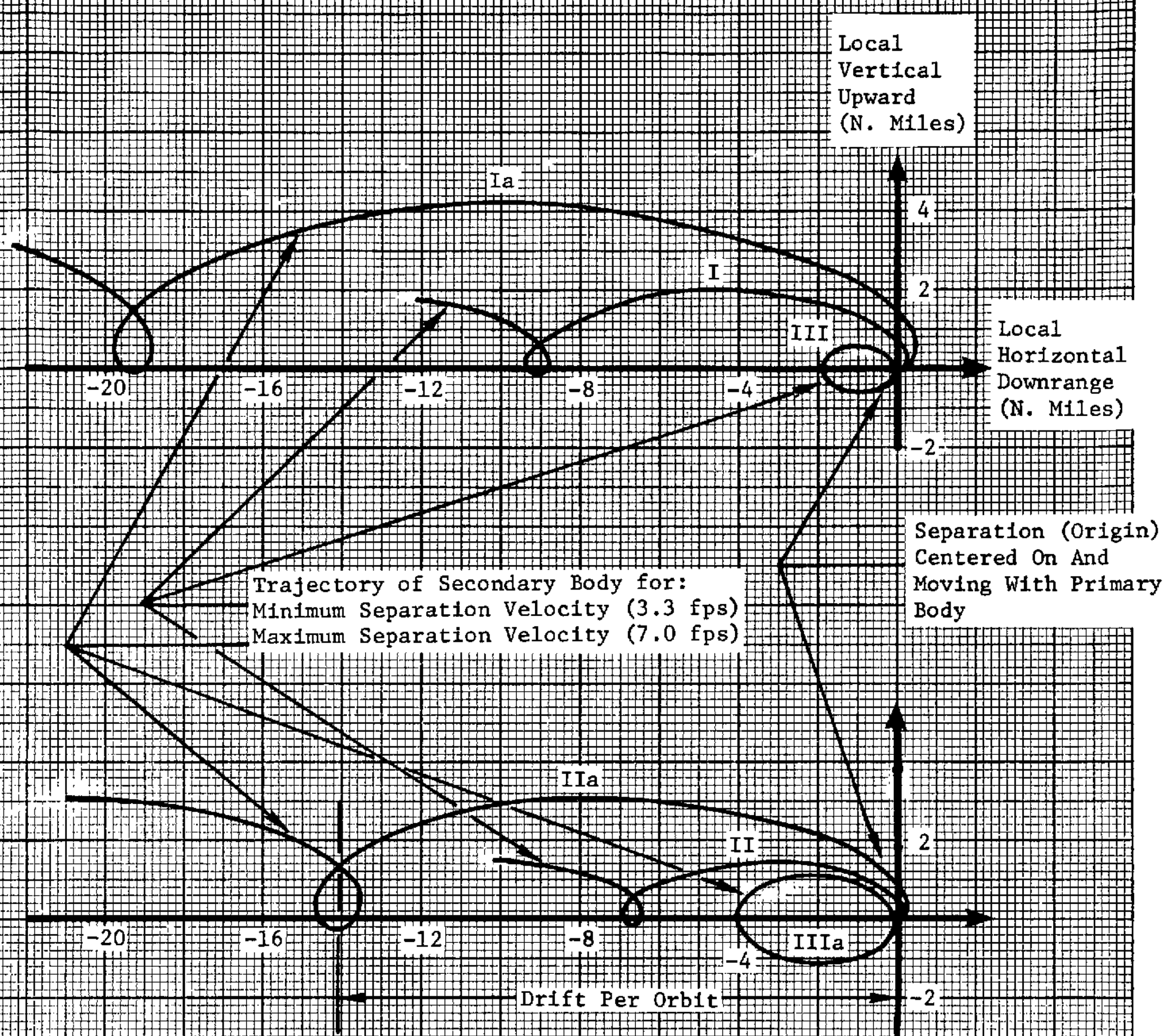
```
INOP16(1:NEW VECTORS AND NEW MANEUVERS,2:NEW MANEUVERS ONLY,3:STOP)
```

FIGURE 1 - MANEUVER IIa

```
MANEUVER TABLE
BODY(1 OR 2),PITCH(DEG),YAW(DEG),DELTA V(FPS),COAST(SEC),NO.OF POINTS)
B,P,Y,DV,CT,NP FOR PHASE 1
? 2
? 45.
? 0.
? 7.0
? 10000.
? 10.
     T        RPDR       RPRA       RPCR        RVDR        RVRA       RVCR
   (SEC)      (NM)       (NM)       (NM)       (FPS)       (FPS)      (FPS)
     0        0.0        0.0        0.0        0.000       0.000      0.000
PHASE NUMBER   1
     0        0.0        0.0        0.0        4.950       4.950      0.000
  1000        -.7        1.5        0.0      -15.198      11.066      0.000
  2000       -5.0        2.9        0.0      -34.940       4.658      0.000
  3000      -10.8        2.7        0.0      -31.972      -7.012      0.000
  4000      -14.4        1.1        0.0       -9.637     -10.753      0.000
  5000      -14.3        -.1        0.0        6.798      -2.313      0.000
  6000      -13.5         .5        0.0       -1.302       8.754      0.000
  7000      -15.7        2.2        0.0      -24.749       9.906      0.000
  8000      -21.0        3.1        0.0      -36.993       -.140      0.000
  9000      -26.4        2.1        0.0      -24.198     -10.022      0.000
 10000      -28.4         .4        0.0        -.824      -8.560      0.000
INOPT6(1:NEW VECTORS AND NEW MANEUVERS,2:NEW MANEUVERS ONLY,3:STOP)
```

FIGURE 1 - MANEUVER IIIa

```
MANEUVER TABLE
BODY(1 OR 2),PITCH(DEG),YAW(DEG),DELTA V(FPS),COAST(SEC),NO.OF POINTS)
B,P,Y,DV,CT,NP FOR PHASE 1
? 2
? 90.
? 0.
? 7.0
? 10000.
? 10.
     T        RPDR       RPRA       RPCR        RVDR        RVRA       RVCR
   (SEC)      (NM)       (NM)       (NM)       (FPS)       (FPS)      (FPS)
     0        0.0        0.0        0.0        0.000       0.000      0.000
PHASE NUMBER   1
     0        0.0        0.0        0.0         .000       7.000      0.000
  1000       -1.2         .9        0.0      -12.617       3.036      0.000
  2000       -3.3         .8        0.0      -10.944      -4.365      0.000
  3000       -4.1        -.2        0.0        3.124      -6.824      0.000
  4000       -2.5       -1.0        0.0       13.652      -1.546      0.000
  5000        -.4        -.6        0.0        8.696       5.495      0.000
  6000        -.2         .4        0.0       -6.120       6.296      0.000
  7000       -2.1        1.0        0.0      -14.002       -.025      0.000
  8000       -3.9         .4        0.0       -6.028      -6.318      0.000
  9000       -3.7        -.6        0.0        8.776      -5.454      0.000
 10000       -1.6       -1.0        0.0       13.630       1.595      0.000
INPUT6(1:NEW VECTORS AND NEW MANEUVERS,2:NEW MANEUVERS ONLY,3:STOP)
```



Maneuver Table

Maneuver	Pitch	Yaw
I & Ia	0°	0°
II & IIa	45°	0°
III & IIIa	90°	0°

Figure 1. Motion of a Secondary Body Relative to a Primary Body - Posigrade Separation Maneuvers (Side Views)

FIGURE 2 - MANEUVER I

```
MANEUVER TABLE
BODY(1 OR 2),PITCH(DEG),YAW(DEG),DELTA V(FPS),COAST(SEC),NO.OF POINTS)
B,P,Y,DV,C1,NP FOR PHASE 1
? 2
? 30.
? 30.
? 3.3
? 10000.
? 10.
```

T (SEC)	RPDR (NM)	RPRA (NM)	RPCR (NM)	RVDR (FPS)	RVRA (FPS)	RVCR (FPS)
0	0.0	0.0	0.0	0.000	0.000	0.000
PHASE NUMBER	1					
0	0.0	0.0	0.0	2.475	1.429	1.650
1000	-.2	.6	.2	-5.710	5.080	.715
2000	-2.0	1.3	.2	-15.838	2.977	-1.029
3000	-4.8	1.4	-.1	-16.441	-2.495	-1.608
4000	-6.8	.7	-.2	-6.837	-5.143	-.366
5000	-7.1	.0	-.2	2.191	-1.964	1.291
6000	-6.7	.2	.1	.243	3.442	1.485
7000	-7.5	.9	.2	-10.304	4.948	-.003
8000	-9.9	1.5	.1	-17.591	.353	-1.487
9000	-12.7	1.2	-.2	-13.369	-4.296	-1.237
10000	-14.0	.4	-.2	-2.416	-4.502	.371

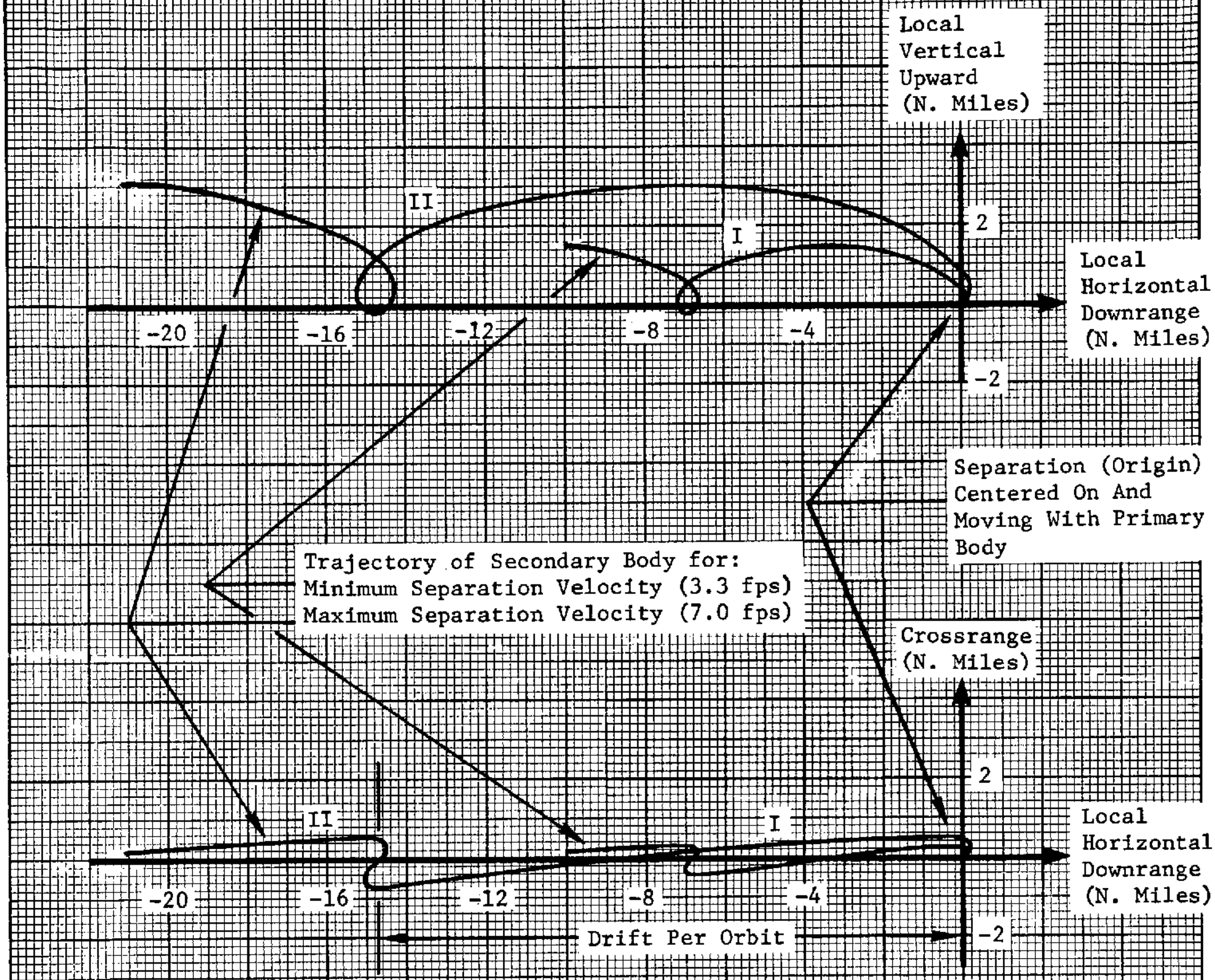
```
INOP16(1:NEW VECTORS AND NEW MANEUVERS,2:NEW MANEUVERS ONLY,3:STOP)
```

FIGURE 2 - MANEUVER II

```
MANEUVER TABLE
BODY(1 OR 2),PITCH(DEG),YAW(DEG),DELTA V(FPS),COAST(SEC),NO.OF POINTS)
B,P,Y,DV,C1,NP FOR PHASE 1
? 2
? 30.
? 30.
? 7.0
? 10000.
? 10.
```

T (SEC)	RPDR (NM)	RPRA (NM)	RPCR (NM)	RVDR (FPS)	RVRA (FPS)	RVCR (FPS)
0	0.0	0.0	0.0	0.000	0.000	0.000
PHASE NUMBER	1					
0	0.0	0.0	0.0	5.250	3.031	3.500
1000	-.3	1.3	.5	-12.111	10.776	1.518
2000	-4.2	2.8	.4	-33.592	6.321	-2.180
3000	-10.2	2.9	-.1	-34.895	-5.276	-3.410
4000	-14.5	1.5	-.5	-14.549	-10.908	-.782
5000	-15.1	.1	-.3	4.438	-4.188	2.734
6000	-14.3	.3	.2	.546	7.284	3.154
7000	-16.0	2.0	.5	-21.809	10.506	.002
8000	-21.1	3.1	.2	-37.308	1.849	-3.149
9000	-26.8	2.5	-.3	-28.432	-8.890	-2.738
10000	-29.6	.8	-.5	-5.209	-9.569	.773

```
INOP16(1:NEW VECTORS AND NEW MANEUVERS,2:NEW MANEUVERS ONLY,3:STOP)
```

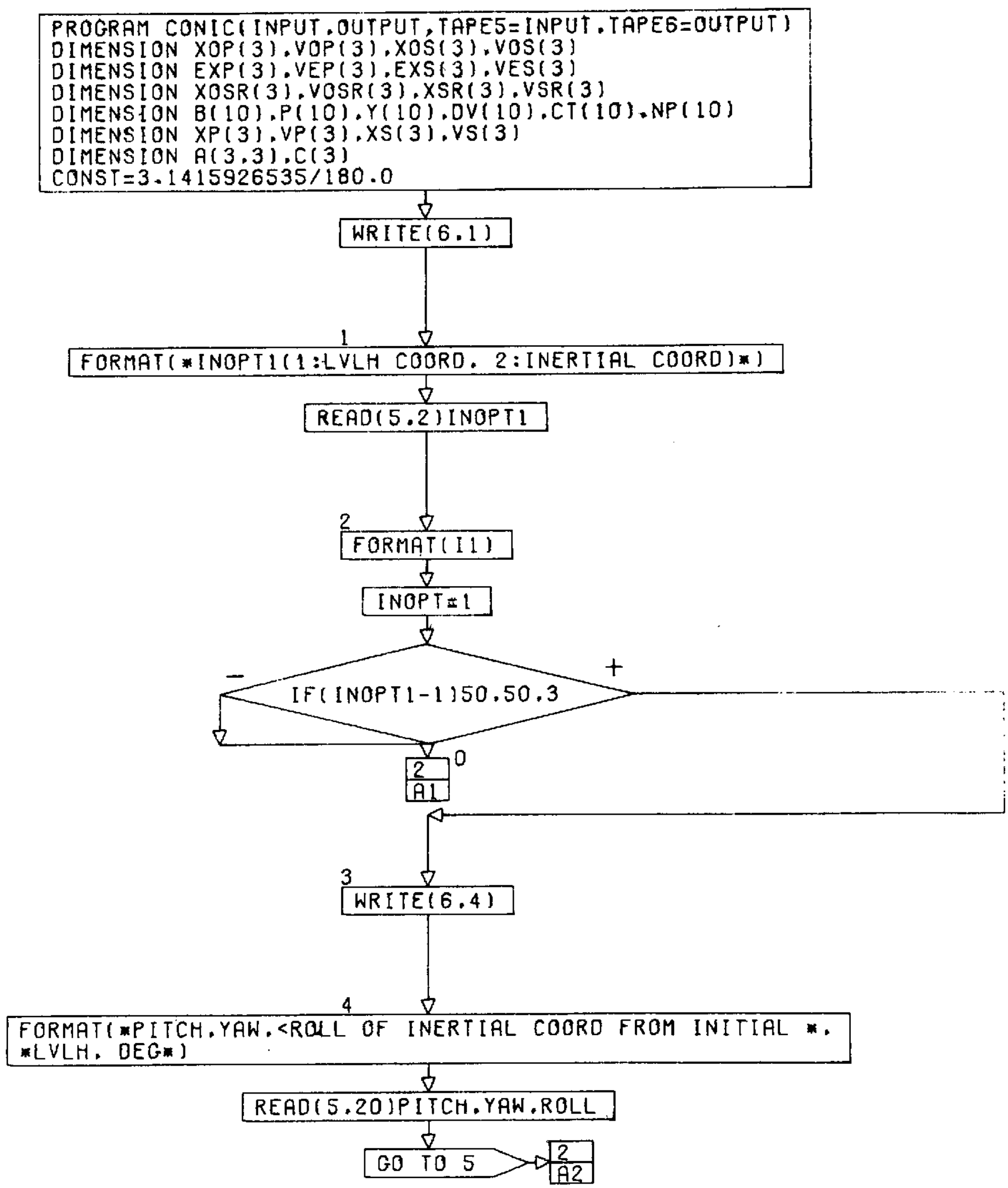


Maneuver Table

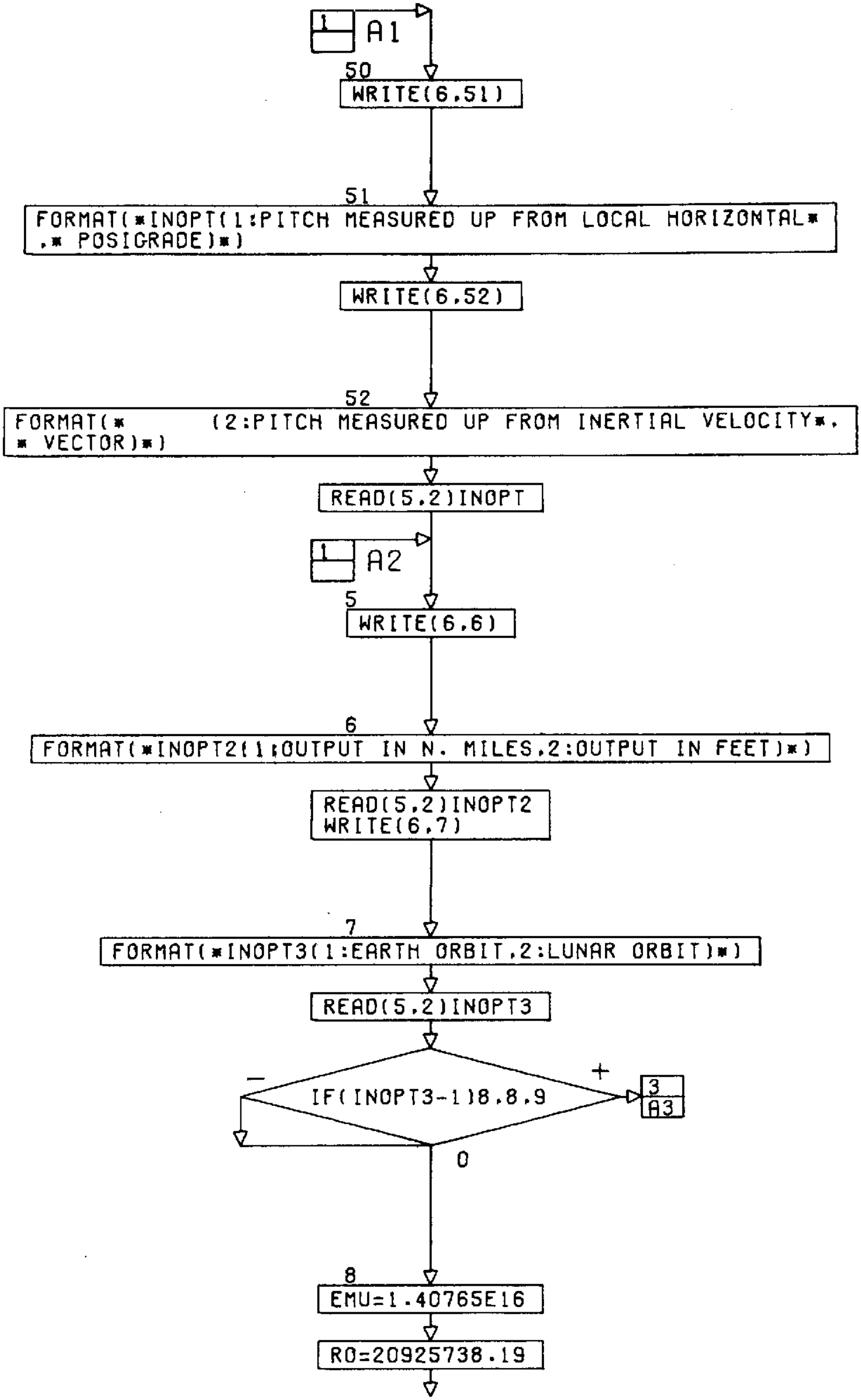
Maneuver	Pitch	Yaw
I	30°	30°
II	30°	30°

Figure 2. Motion of a Secondary Body Relative to a Primary Body - Posigrade Separation Maneuvers (Side and Top Views)

6.0 DETAILED PROGRAM FLOW CHART

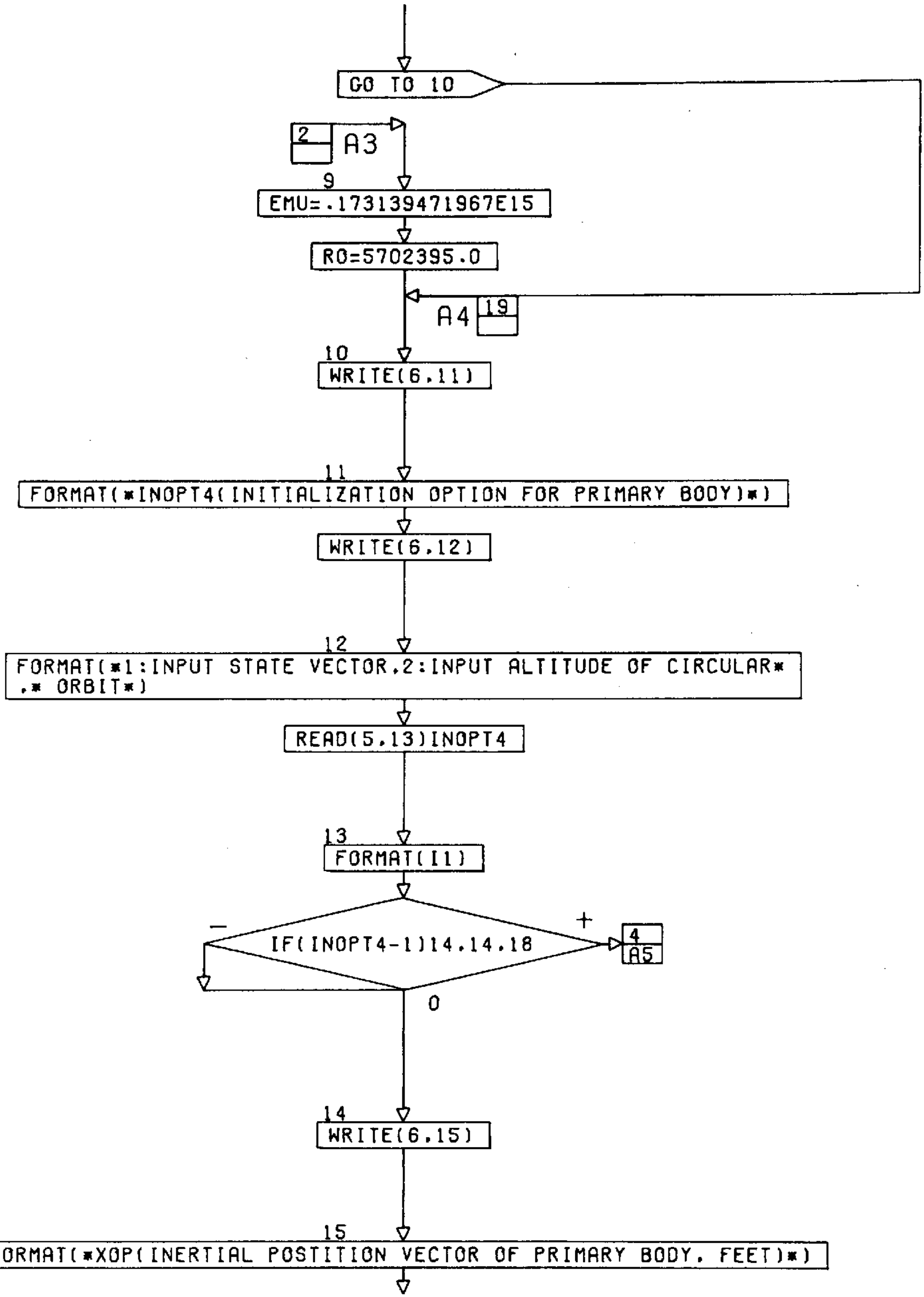


CONT. ON PG 2



1
A1
50
WRITE(6,51)
51
FORMAT(*INOPT(1:PITCH MEASURED UP FROM LOCAL HORIZONTAL*
,* POSIGRADE)*)
WRITE(6,52)
52
FORMAT(*   (2:PITCH MEASURED UP FROM INERTIAL VELOCITY*,
* VECTOR)*)
READ(5,2)INOPT
1
A2
5
WRITE(6,6)
6
FORMAT(*INOPT2(1:OUTPUT IN N. MILES,2:OUTPUT IN FEET)*)
READ(5,2)INOPT2
WRITE(6,7)
7
FORMAT(*INOPT3(1:EARTH ORBIT,2:LUNAR ORBIT)*)
READ(5,2)INOPT3
−
IF(INOPT3-1)8,8,9
+
3
A3
0
8
EMU=1.40765E16
RO=20925738.19

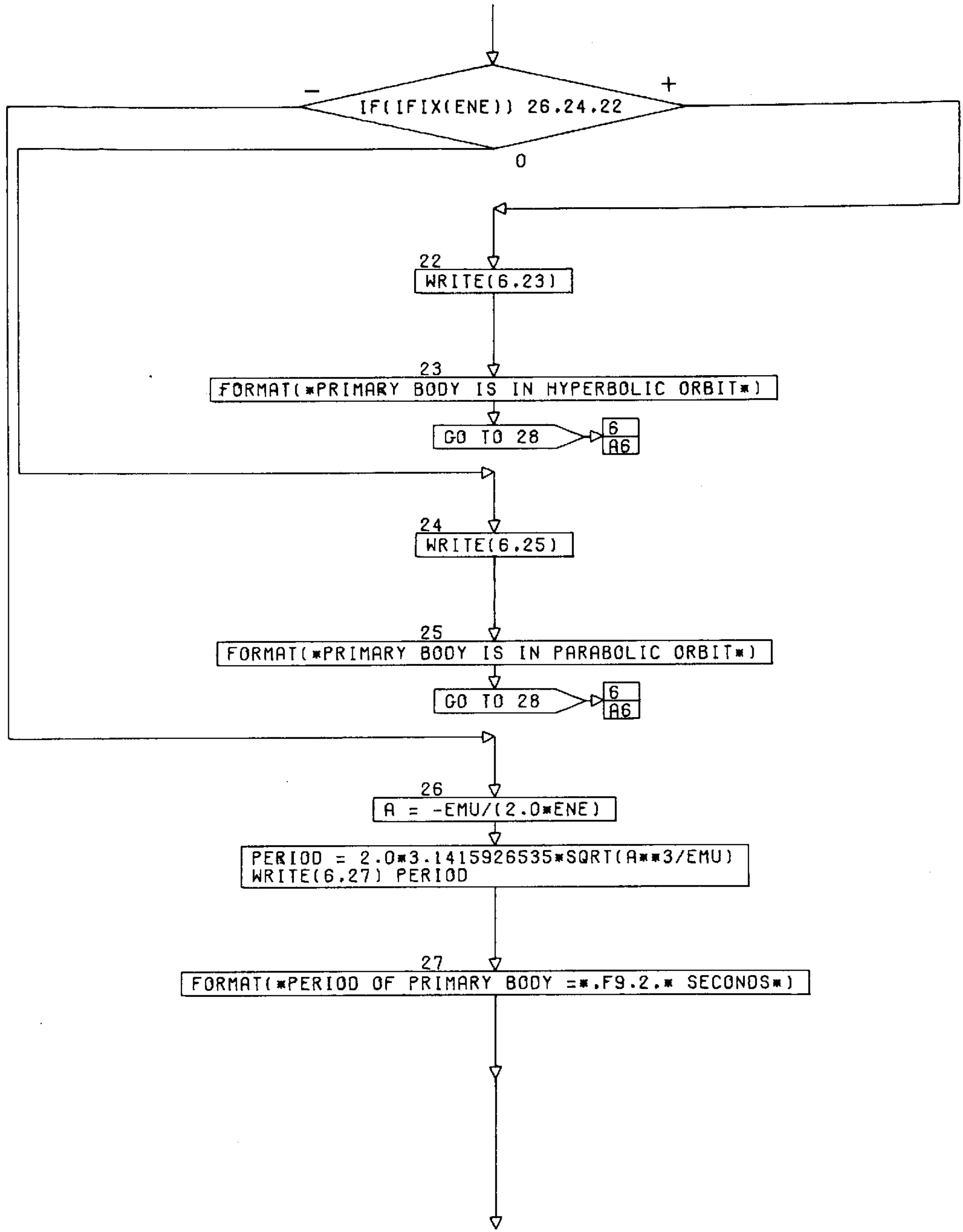
CONT. ON PG 3


GO TO 10
2
A3
9
EMU=.173139471967E15
RO=5702395.0
A4
19
10
WRITE(6,11)
11
FORMAT(*INOPT4(INITIALIZATION OPTION FOR PRIMARY BODY)*)
WRITE(6,12)
12
FORMAT(*1:INPUT STATE VECTOR,2:INPUT ALTITUDE OF CIRCULAR*
,* ORBIT*)
READ(5,13)INOPT4
13
FORMAT(I1)
IF(INOPT4-1)14,14,18
—
+
0
4
A5
14
WRITE(6,15)
15
FORMAT(*XOP(INERTIAL POSTITION VECTOR OF PRIMARY BODY, FEET)*)

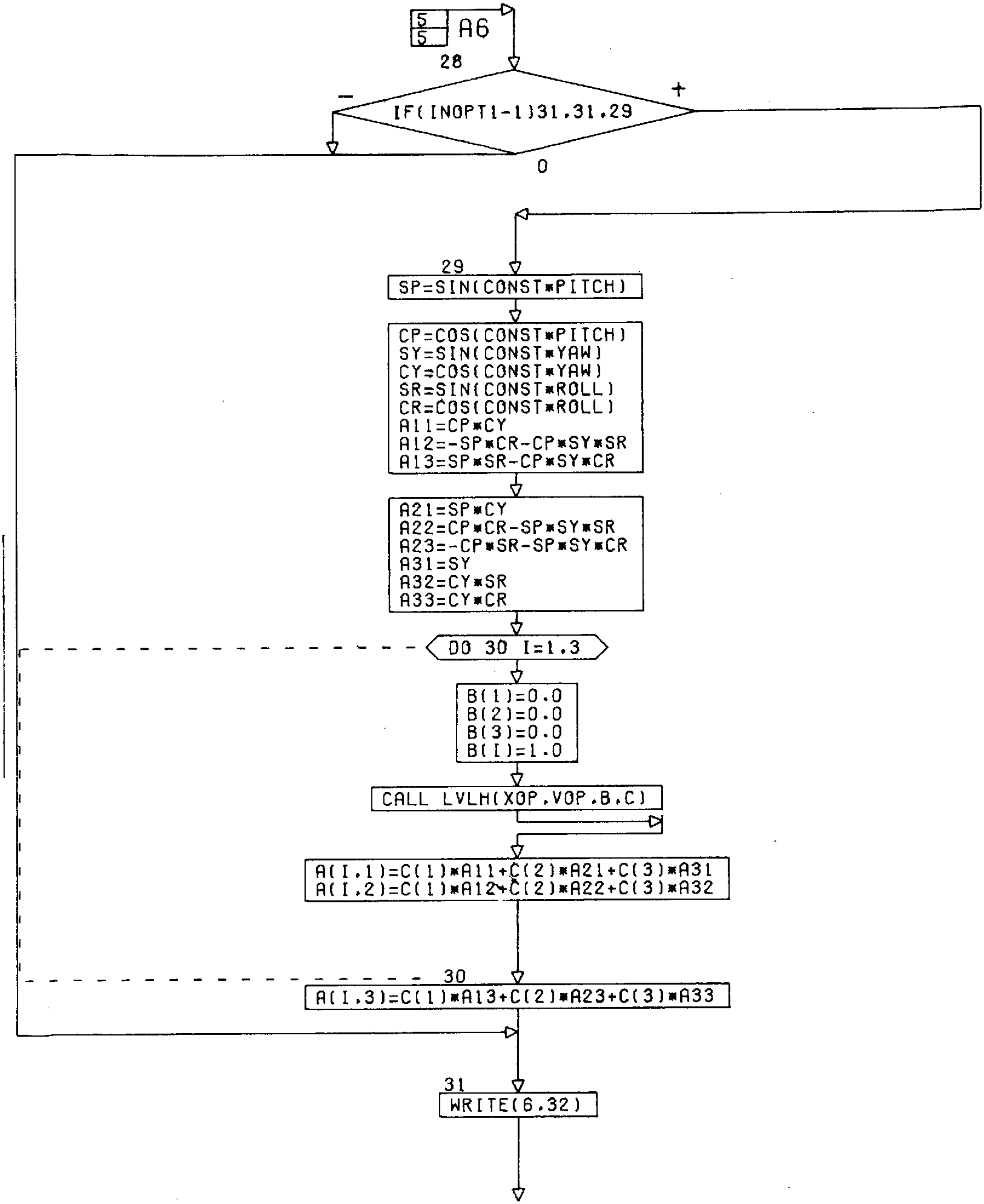
CONT. ON PG 4

```
READ(5,16)XOP(1),XOP(2),XOP(3)

16 FORMAT(F16.4)
WRITE(6,17)

17 FORMAT(*VOP(INERTIAL VELOCITY VECTOR OF PRIMARY BODY,FPS)*)
READ(5,16)VOP(1),VOP(2),VOP(3)
GO TO 21

3 A5
18 WRITE(6,19)

19 FORMAT(*H(CIRCULAR ORBIT ALTITUDE OF PRIMARY BODY,N.M.)*)
READ(5,20)H

20 FORMAT(F16.9)
XOP(1)=0.0
XOP(2)=RO+H*6076.1155
XOP(3)=0.0
VOP(1)=SQRT(EMU/XOP(2))
VOP(2)=0.0
VOP(3)=0.0

21 ARE = SQRT(XOP(1)**2+XOP(2)**2+XOP(3)**2)
VEE = SQRT(VOP(1)**2+VOP(2)**2+VOP(3)**2)
ENE = 0.5*VEE**2 - EMU/ARE
```

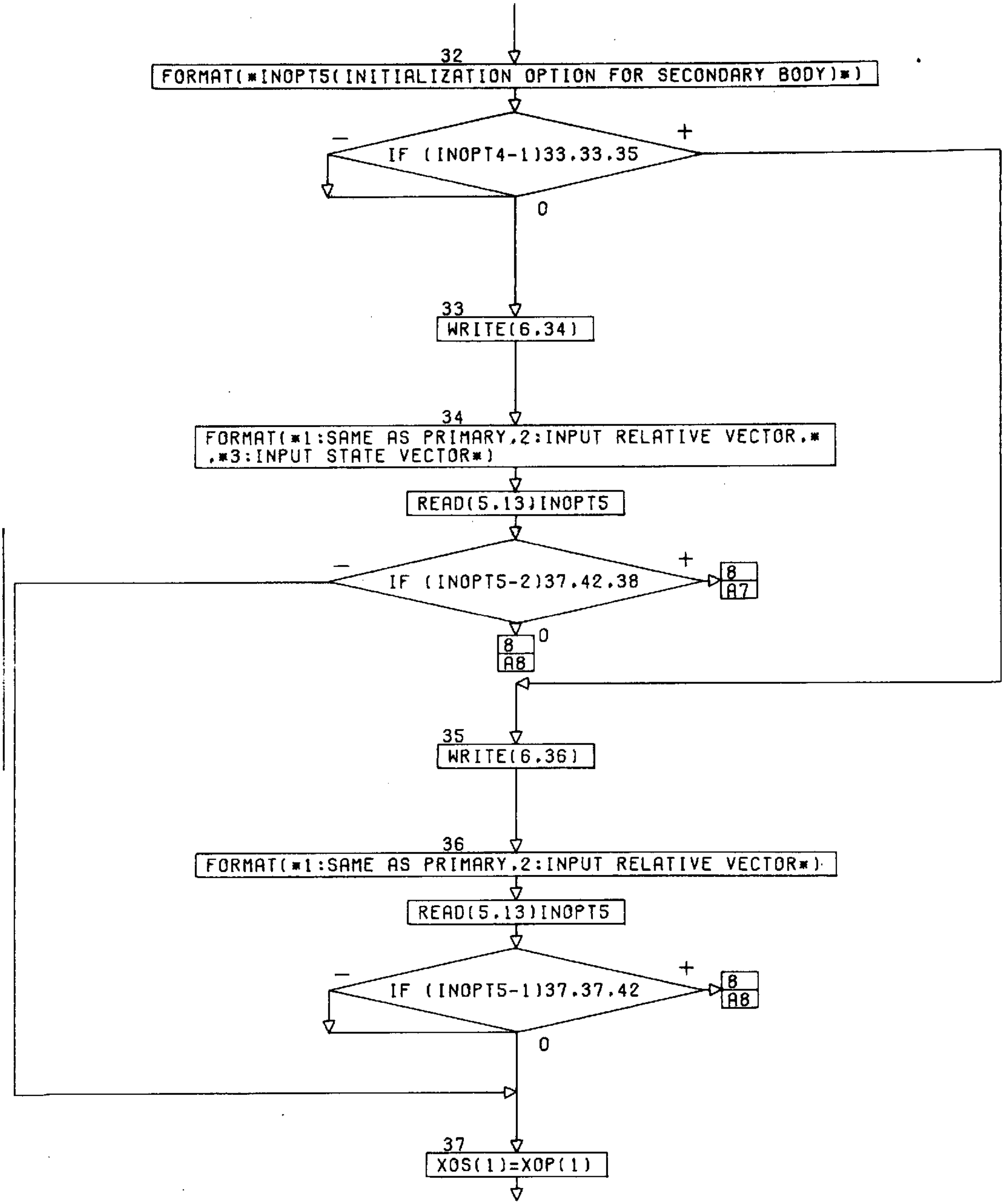
CONT. ON PG 5


IF(IFIX(ENE)) 26.24.22
−
+
0
22
WRITE(6.23)
23
FORMAT(*PRIMARY BODY IS IN HYPERBOLIC ORBIT*)
GO TO 28
6
A6
24
WRITE(6.25)
25
FORMAT(*PRIMARY BODY IS IN PARABOLIC ORBIT*)
GO TO 28
6
A6
26
A = -EMU/(2.0*ENE)
PERIOD = 2.0*3.1415926535*SQRT(A**3/EMU)
WRITE(6.27) PERIOD
27
FORMAT(*PERIOD OF PRIMARY BODY =*.F9.2.* SECONDS*)

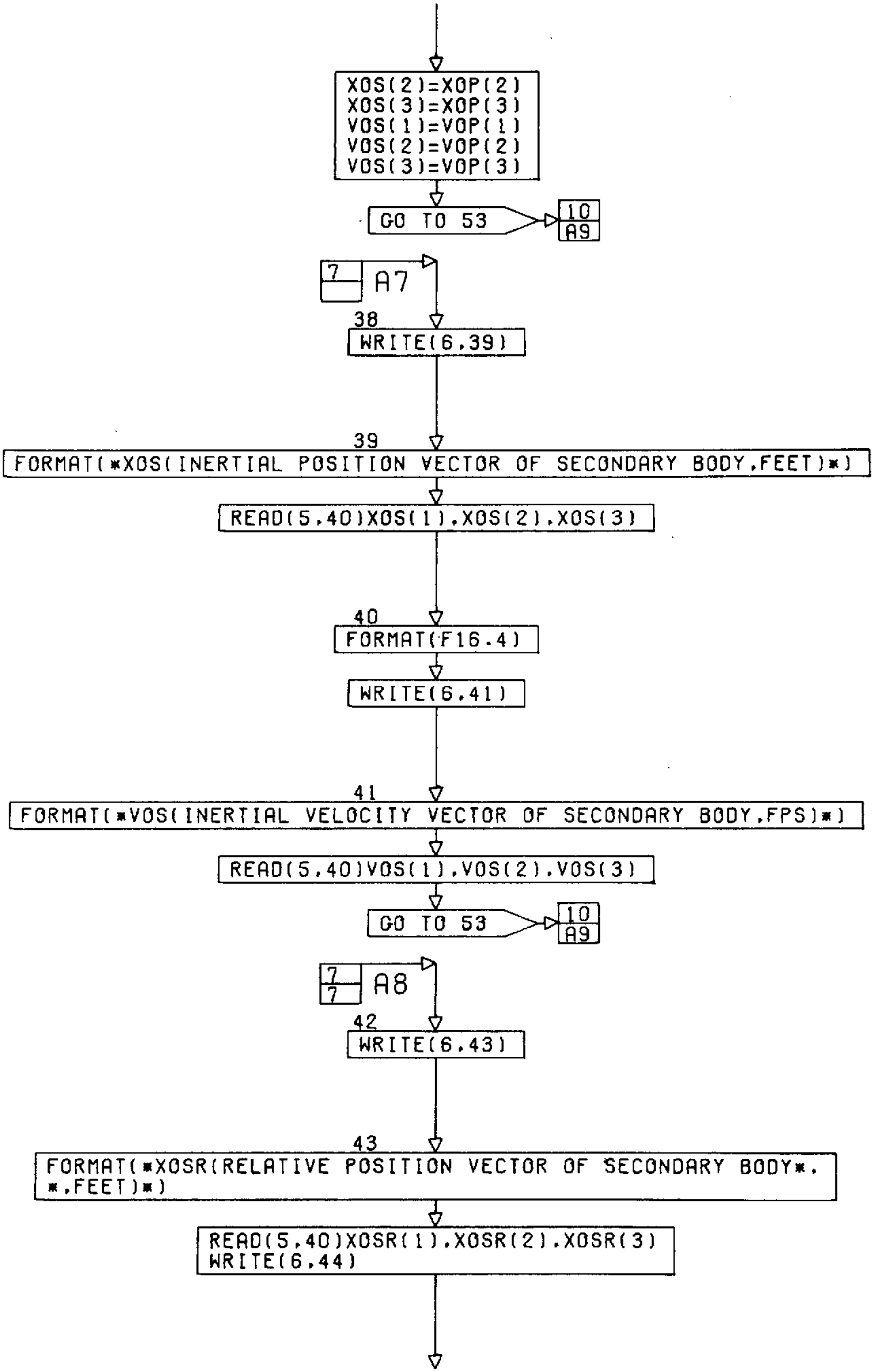
CONT. ON PG 6


5
5
A6
28
IF(INOPT1-1)31,31,29
−
+
0
29
SP=SIN(CONST*PITCH)
CP=COS(CONST*PITCH)
SY=SIN(CONST*YAW)
CY=COS(CONST*YAW)
SR=SIN(CONST*ROLL)
CR=COS(CONST*ROLL)
A11=CP*CY
A12=-SP*CR-CP*SY*SR
A13=SP*SR-CP*SY*CR
A21=SP*CY
A22=CP*CR-SP*SY*SR
A23=-CP*SR-SP*SY*CR
A31=SY
A32=CY*SR
A33=CY*CR
DO 30 I=1,3
B(1)=0.0
B(2)=0.0
B(3)=0.0
B(I)=1.0
CALL LVLH(XOP,VOP,B,C)
A(I,1)=C(1)*A11+C(2)*A21+C(3)*A31
A(I,2)=C(1)*A12+C(2)*A22+C(3)*A32
30
A(I,3)=C(1)*A13+C(2)*A23+C(3)*A33
31
WRITE(6,32)

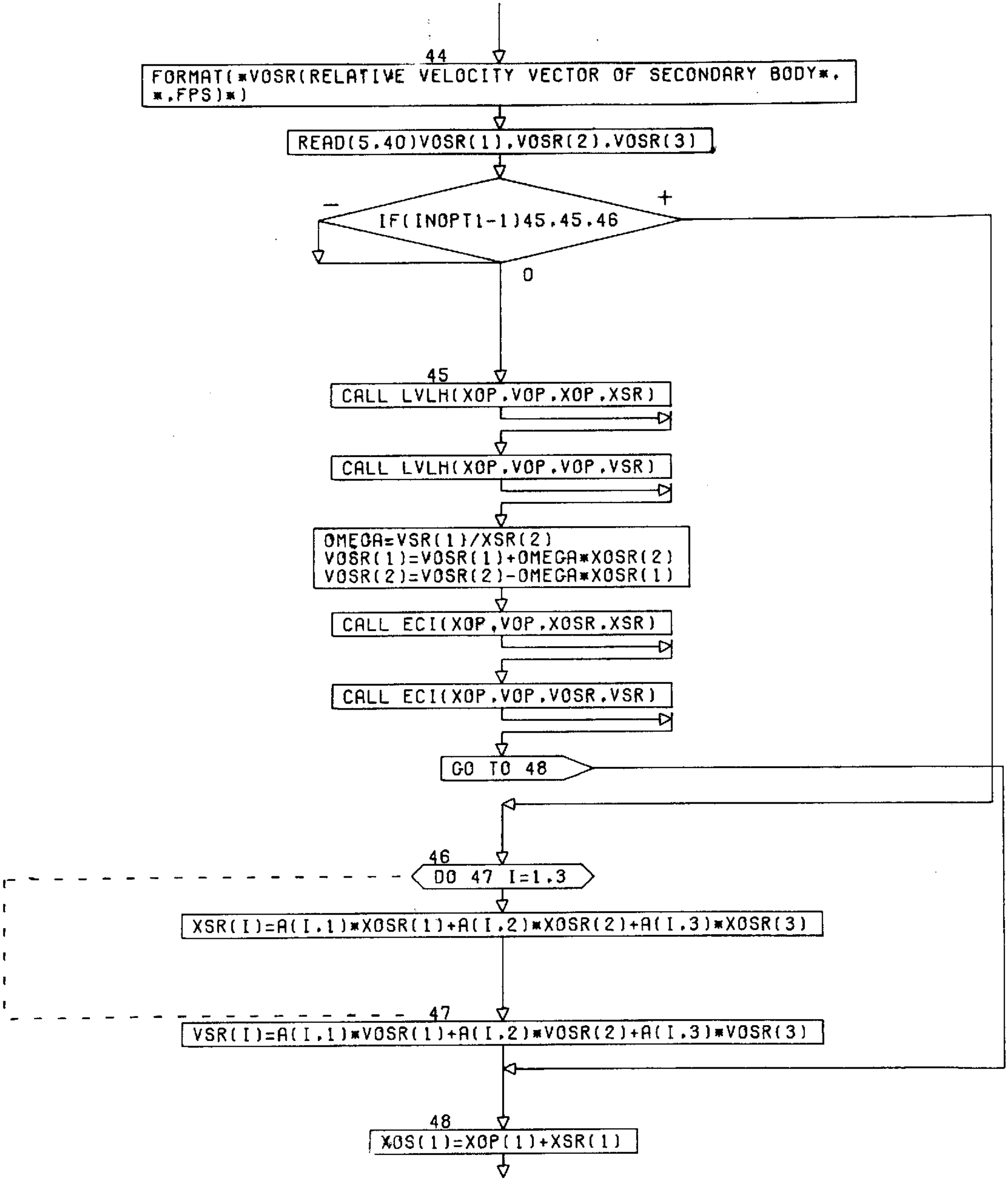
CONT. ON PG 7


32
FORMAT(*INOPT5(INITIALIZATION OPTION FOR SECONDARY BODY)*)
IF (INOPT4-1)33,33,35
−
+
0
33
WRITE(6,34)
34
FORMAT(*1:SAME AS PRIMARY,2:INPUT RELATIVE VECTOR,* ,*3:INPUT STATE VECTOR*)
READ(5,13)INOPT5
IF (INOPT5-2)37,42,38
−
+
8
A7
0
8
A8
35
WRITE(6,36)
36
FORMAT(*1:SAME AS PRIMARY,2:INPUT RELATIVE VECTOR*)
READ(5,13)INOPT5
IF (INOPT5-1)37,37,42
−
+
8
A8
0
37
XOS(1)=XOP(1)

CONT. ON PG 8


XOS(2)=XOP(2)
XOS(3)=XOP(3)
VOS(1)=VOP(1)
VOS(2)=VOP(2)
VOS(3)=VOP(3)
GO TO 53
10
A9
7
A7
38
WRITE(6,39)
39
FORMAT(*XOS(INERTIAL POSITION VECTOR OF SECONDARY BODY,FEET)*)
READ(5,40)XOS(1),XOS(2),XOS(3)
40
FORMAT(F16.4)
WRITE(6,41)
41
FORMAT(*VOS(INERTIAL VELOCITY VECTOR OF SECONDARY BODY,FPS)*)
READ(5,40)VOS(1),VOS(2),VOS(3)
GO TO 53
10
A9
7
7
A8
42
WRITE(6,43)
43
FORMAT(*XOSR(RELATIVE POSITION VECTOR OF SECONDARY BODY*,
,FEET))
READ(5,40)XOSR(1),XOSR(2),XOSR(3)
WRITE(6,44)

CONT. ON PG 9


44
FORMAT(*VOSR(RELATIVE VELOCITY VECTOR OF SECONDARY BODY*,
,FPS))
READ(5,40)VOSR(1),VOSR(2),VOSR(3)
IF(INOPT1-1)45,45,46
-
+
0
45
CALL LVLH(XOP,VOP,XOP,XSR)
CALL LVLH(XOP,VOP,VOP,VSR)
OMEGA=VSR(1)/XSR(2)
VOSR(1)=VOSR(1)+OMEGA*XOSR(2)
VOSR(2)=VOSR(2)-OMEGA*XOSR(1)
CALL ECI(XOP,VOP,XOSR,XSR)
CALL ECI(XOP,VOP,VOSR,VSR)
GO TO 48
46
DO 47 I=1,3
XSR(I)=A(I,1)*XOSR(1)+A(I,2)*XOSR(2)+A(I,3)*XOSR(3)
47
VSR(I)=A(I,1)*VOSR(1)+A(I,2)*VOSR(2)+A(I,3)*VOSR(3)
48
XOS(1)=XOP(1)+XSR(1)

CONT. ON PG 10

XOS(2)=XOP(2)+XSR(2)
XOS(3)=XOP(3)+XSR(3)
VOS(1)=VOP(1)+VSR(1)
VOS(2)=VOP(2)+VSR(2)
VOS(3)=VOP(3)+VSR(3)

8 / 8 A9   A9 19

53 WRITE(6,54)

54 FORMAT(*NPHASE(NUMBER OF MANEUVER PHASES, MAX OF 10)*,
(FORMAT I3))

READ(5,60)NPHASE
WRITE(6,55)

55 FORMAT(*MANEUVER TABLE*)

WRITE(6,56)

56 FORMAT(*BODY(1 OR 2),PITCH(DEG),YAW(DEG),DELTA V(FPS),COAST*,
(SEC),NO.OF POINTS))

DO 59 I=1,NPHASE - - - ▷ 11

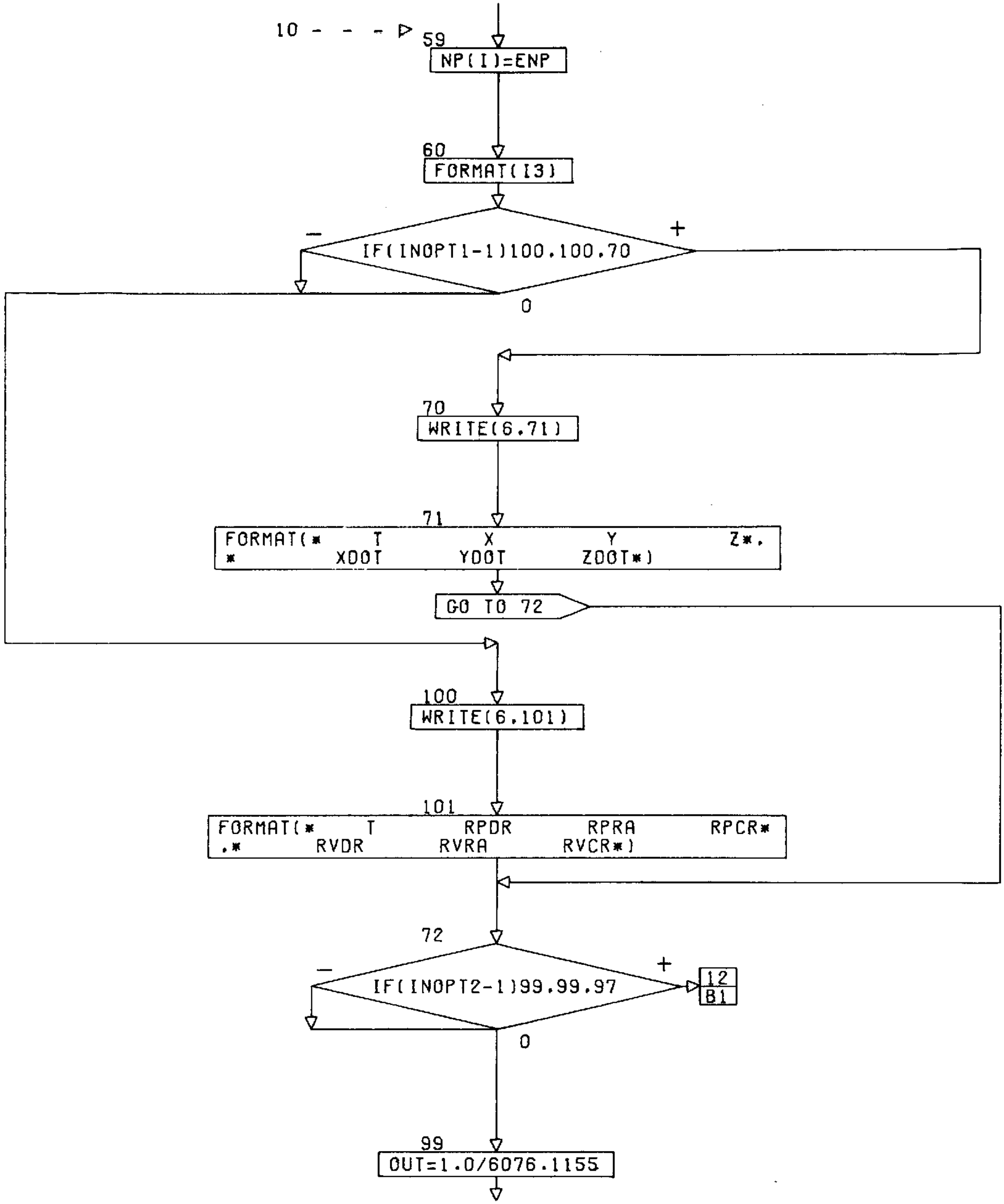
WRITE(6,57)I

57 FORMAT(*B,P,Y,DV,CT,NP FOR PHASE*,I2)

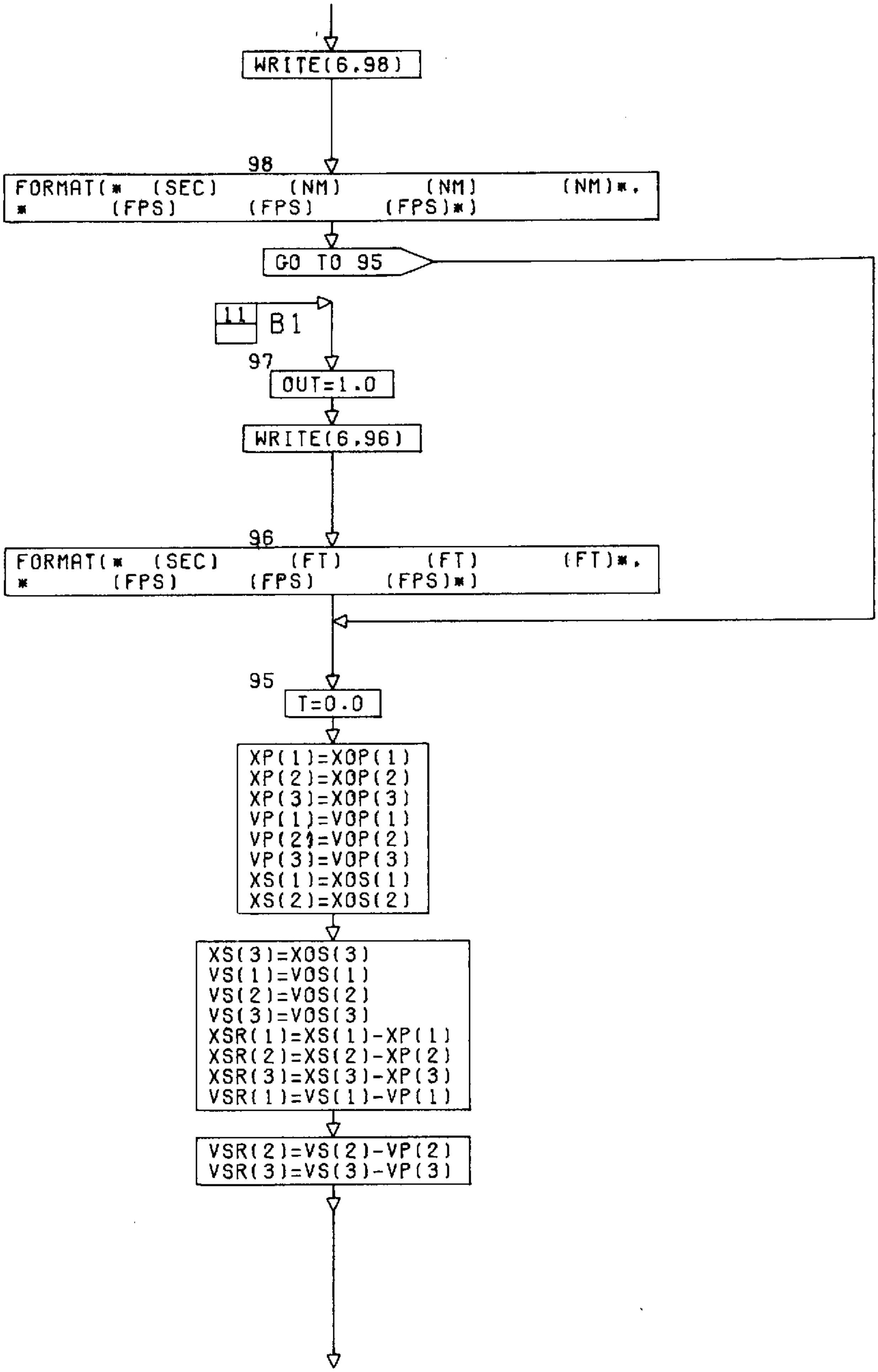
READ(5,58)B(I),P(I),Y(I),DV(I),CT(I),ENP

58 FORMAT(F16.9)

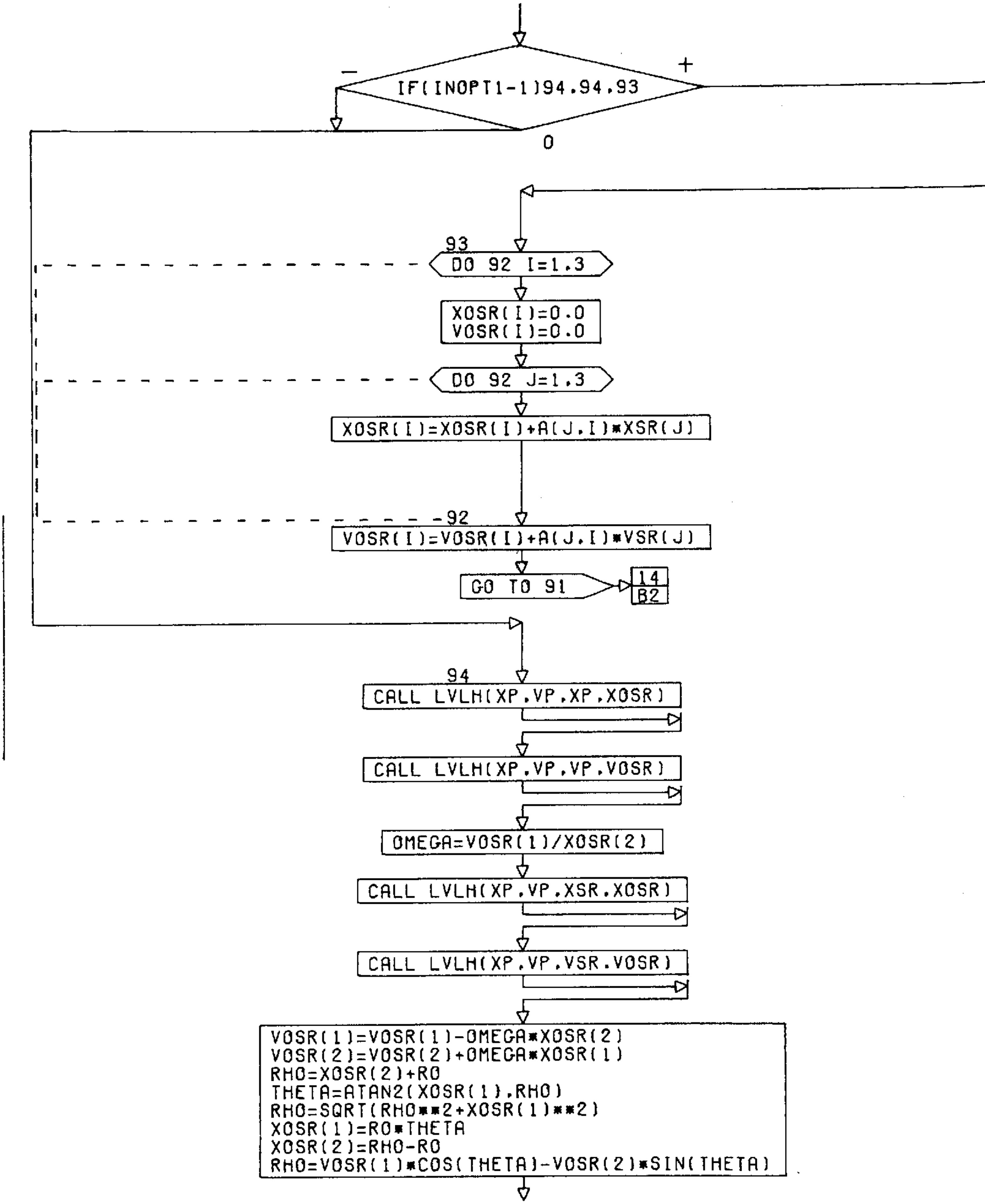
CONT. ON PG 11


10
59
NP(I)=ENP
60
FORMAT(I3)
IF(INOPT1-1)100,100,70
-
+
0
70
WRITE(6,71)
71
FORMAT(* T X Y Z*,
* XDOT YDOT ZDOT*)
GO TO 72
100
WRITE(6,101)
101
FORMAT(* T RPDR RPRA RPCR*
,* RVDR RVRA RVCR*)
72
IF(INOPT2-1)99,99,97
-
+
0
12
B1
99
OUT=1.0/6076.1155

CONT. ON PG 12

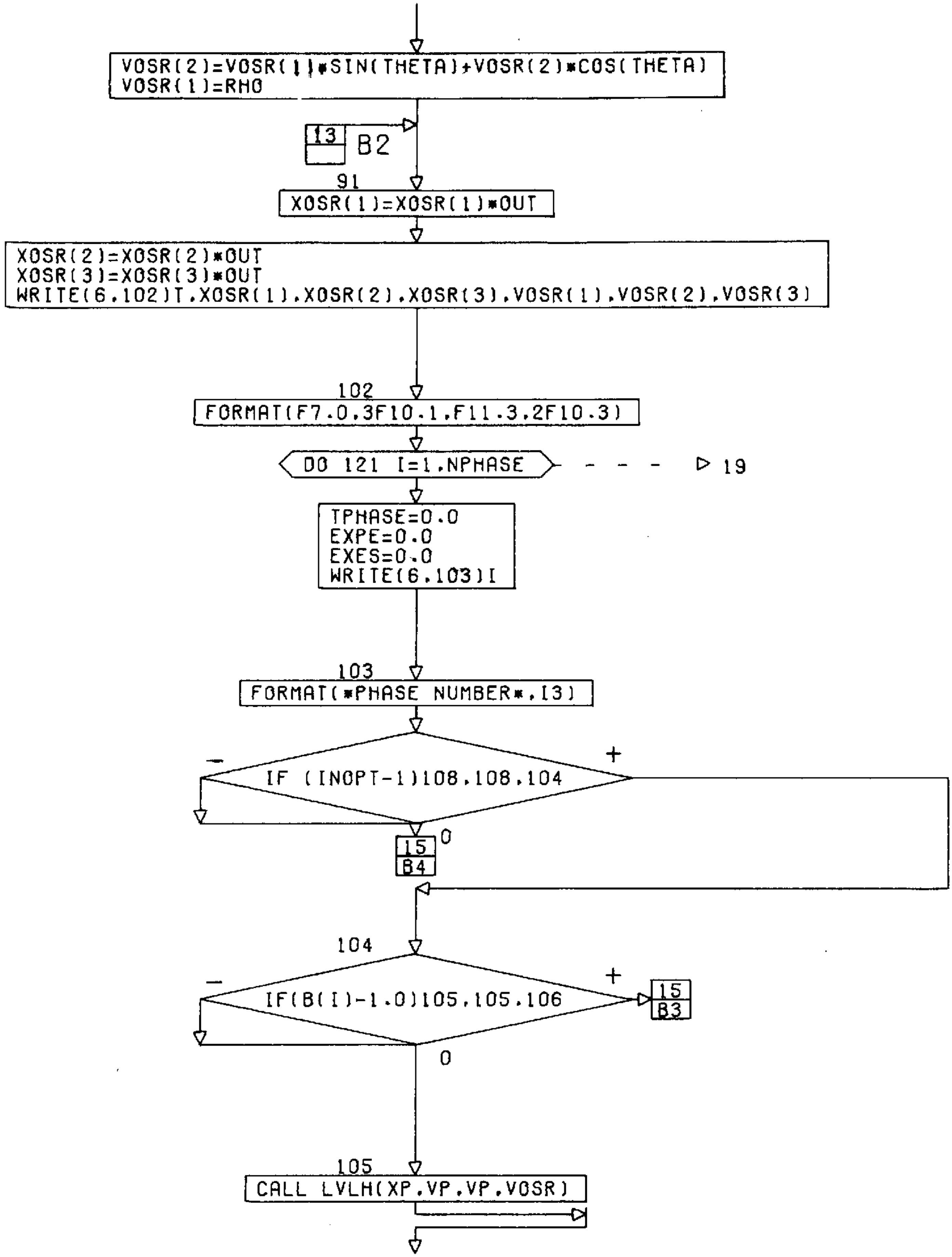

WRITE(6,98)
98
FORMAT(* (SEC) (NM) (NM) (NM)*,
* (FPS) (FPS) (FPS)*)
GO TO 95
11
B1
97
OUT=1.0
WRITE(6,96)
96
FORMAT(* (SEC) (FT) (FT) (FT)*,
* (FPS) (FPS) (FPS)*)
95
T=0.0
XP(1)=XOP(1)
XP(2)=XOP(2)
XP(3)=XOP(3)
VP(1)=VOP(1)
VP(2)=VOP(2)
VP(3)=VOP(3)
XS(1)=XOS(1)
XS(2)=XOS(2)
XS(3)=XOS(3)
VS(1)=VOS(1)
VS(2)=VOS(2)
VS(3)=VOS(3)
XSR(1)=XS(1)-XP(1)
XSR(2)=XS(2)-XP(2)
XSR(3)=XS(3)-XP(3)
VSR(1)=VS(1)-VP(1)
VSR(2)=VS(2)-VP(2)
VSR(3)=VS(3)-VP(3)

CONT. ON PG 13

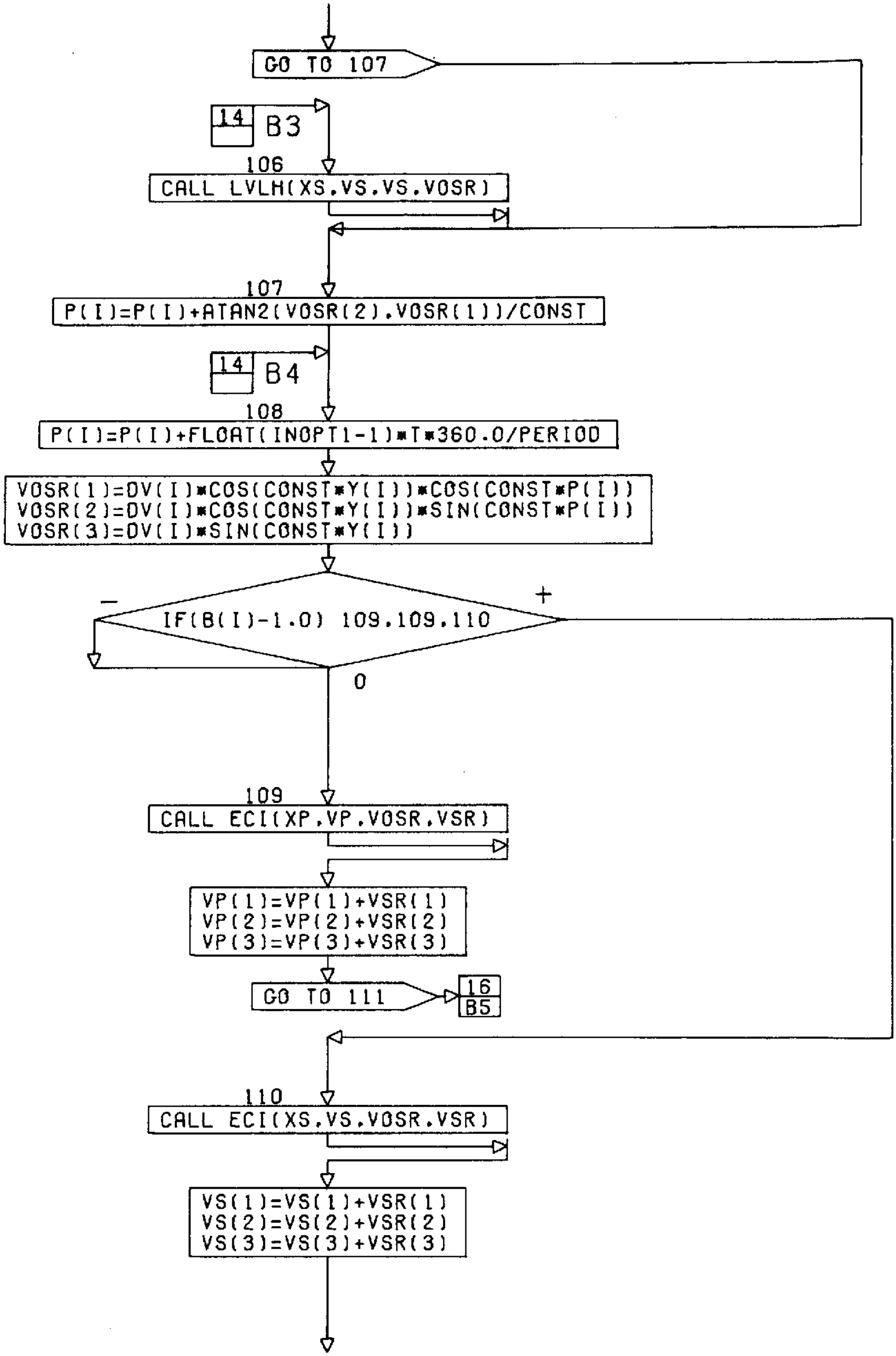


IF(INOPT1-1)94,94,93
−
+
0
93
DO 92 I=1,3
XOSR(I)=0.0
VOSR(I)=0.0
DO 92 J=1,3
XOSR(I)=XOSR(I)+A(J,I)*XSR(J)
-92
VOSR(I)=VOSR(I)+A(J,I)*VSR(J)
GO TO 91
14
B2
94
CALL LVLH(XP,VP,XP,XOSR)
CALL LVLH(XP,VP,VP,VOSR)
OMEGA=VOSR(1)/XOSR(2)
CALL LVLH(XP,VP,XSR,XOSR)
CALL LVLH(XP,VP,VSR,VOSR)
VOSR(1)=VOSR(1)-OMEGA*XOSR(2)
VOSR(2)=VOSR(2)+OMEGA*XOSR(1)
RHO=XOSR(2)+RO
THETA=ATAN2(XOSR(1),RHO)
RHO=SQRT(RHO**2+XOSR(1)**2)
XOSR(1)=RO*THETA
XOSR(2)=RHO-RO
RHO=VOSR(1)*COS(THETA)-VOSR(2)*SIN(THETA)

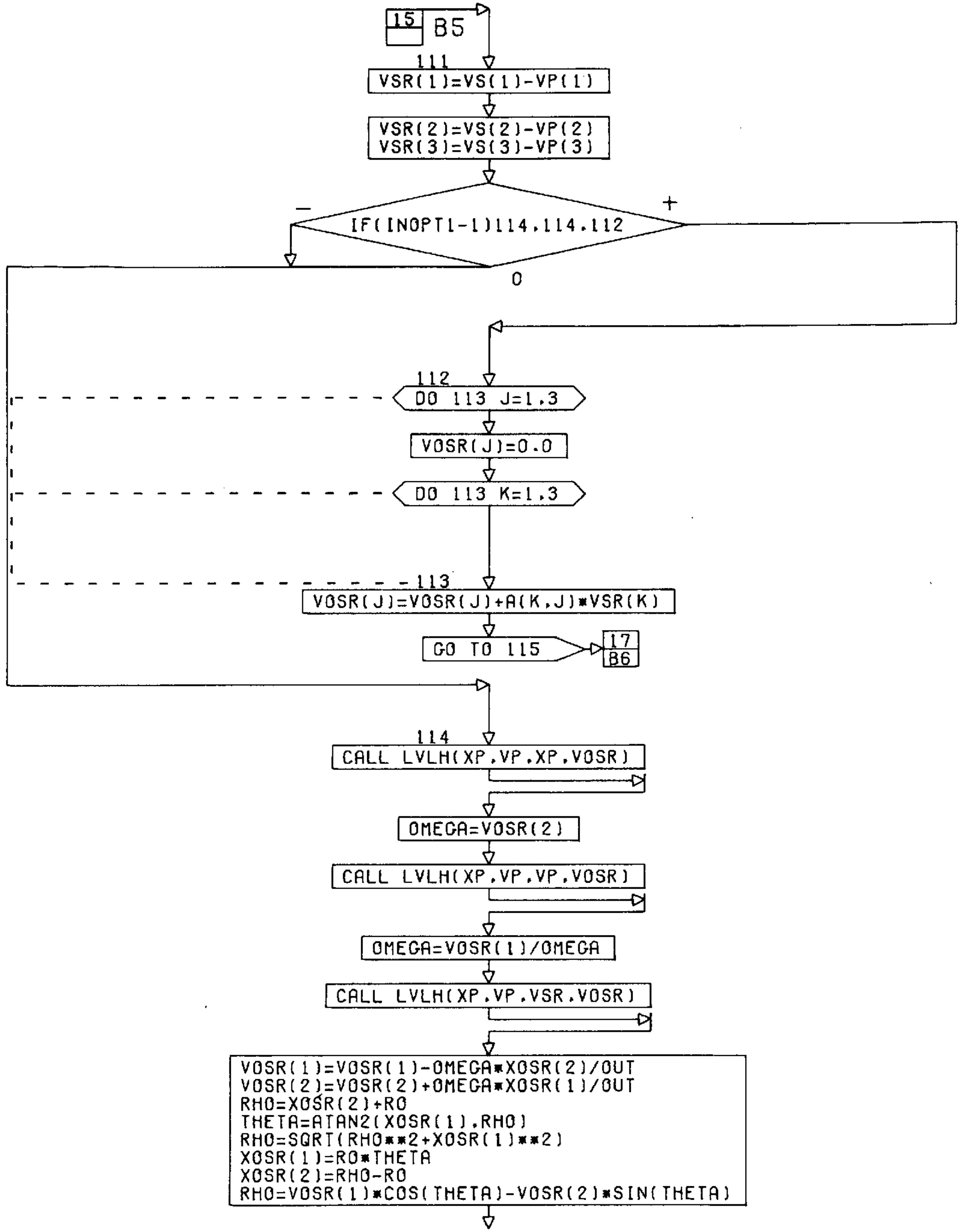
CONT. ON PG 14


VOSR(2)=VOSR(1)*SIN(THETA)+VOSR(2)*COS(THETA)
VOSR(1)=RHO
13
B2
91
XOSR(1)=XOSR(1)*OUT
XOSR(2)=XOSR(2)*OUT
XOSR(3)=XOSR(3)*OUT
WRITE(6,102)T,XOSR(1),XOSR(2),XOSR(3),VOSR(1),VOSR(2),VOSR(3)
102
FORMAT(F7.0,3F10.1,F11.3,2F10.3)
DO 121 I=1,NPHASE
19
TPHASE=0.0
EXPE=0.0
EXES=0.0
WRITE(6,103)I
103
FORMAT(*PHASE NUMBER*,I3)
IF (INOPT-1)108,108,104
−
+
0
15
B4
104
IF(B(I)-1.0)105,105,106
−
+
0
15
B3
105
CALL LVLH(XP,VP,VP,VOSR)

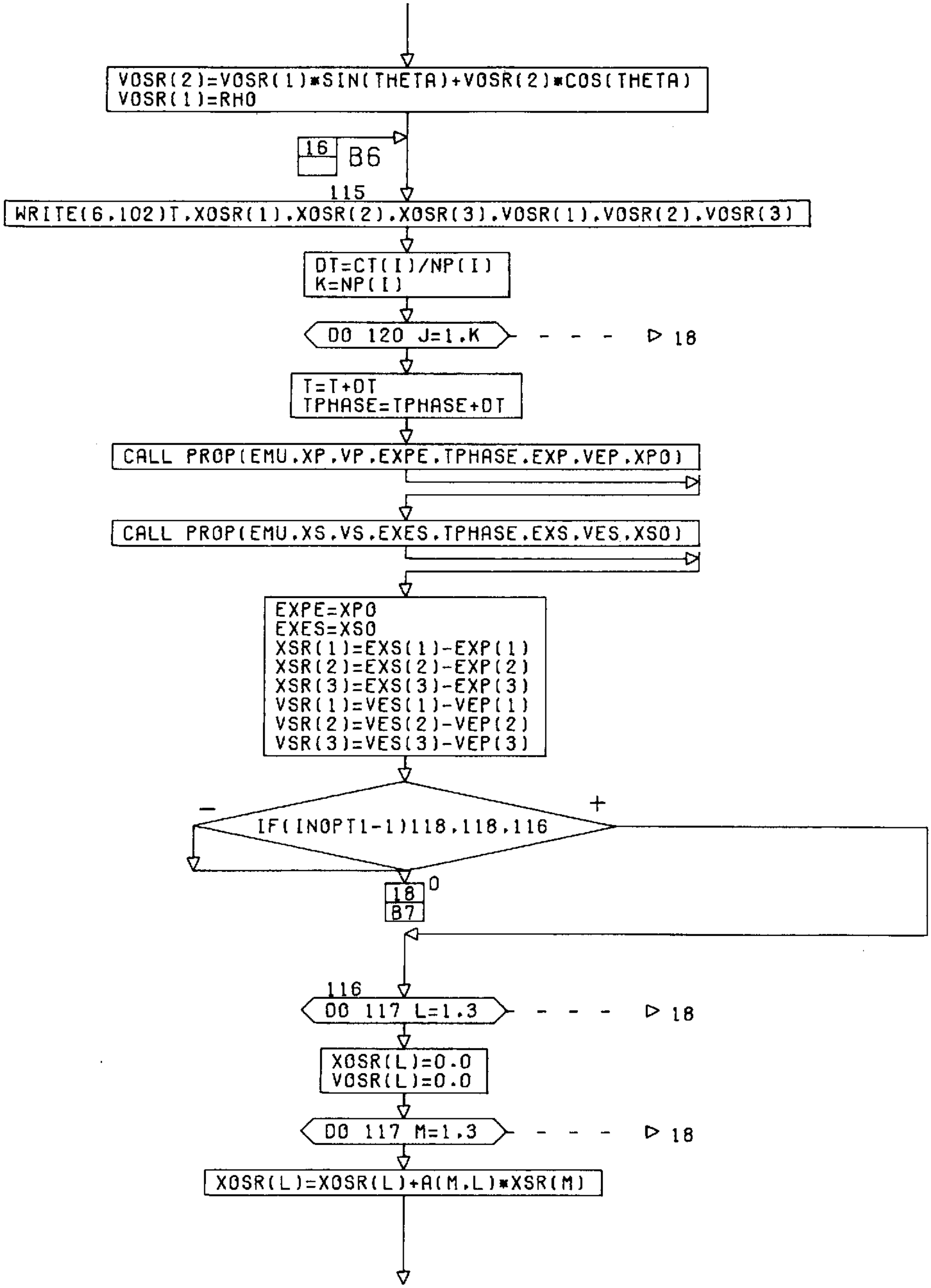
CONT. ON PG 15


GO TO 107
14
B3
106
CALL LVLH(XS.VS.VS.VOSR)
107
P(I)=P(I)+ATAN2(VOSR(2).VOSR(1))/CONST
14
B4
108
P(I)=P(I)+FLOAT(INOPT1-1)*T*360.0/PERIOD
VOSR(1)=DV(I)*COS(CONST*Y(I))*COS(CONST*P(I))
VOSR(2)=DV(I)*COS(CONST*Y(I))*SIN(CONST*P(I))
VOSR(3)=DV(I)*SIN(CONST*Y(I))
−
+
IF(B(I)-1.0) 109.109.110
0
109
CALL ECI(XP.VP.VOSR.VSR)
VP(1)=VP(1)+VSR(1)
VP(2)=VP(2)+VSR(2)
VP(3)=VP(3)+VSR(3)
GO TO 111
16
B5
110
CALL ECI(XS.VS.VOSR.VSR)
VS(1)=VS(1)+VSR(1)
VS(2)=VS(2)+VSR(2)
VS(3)=VS(3)+VSR(3)

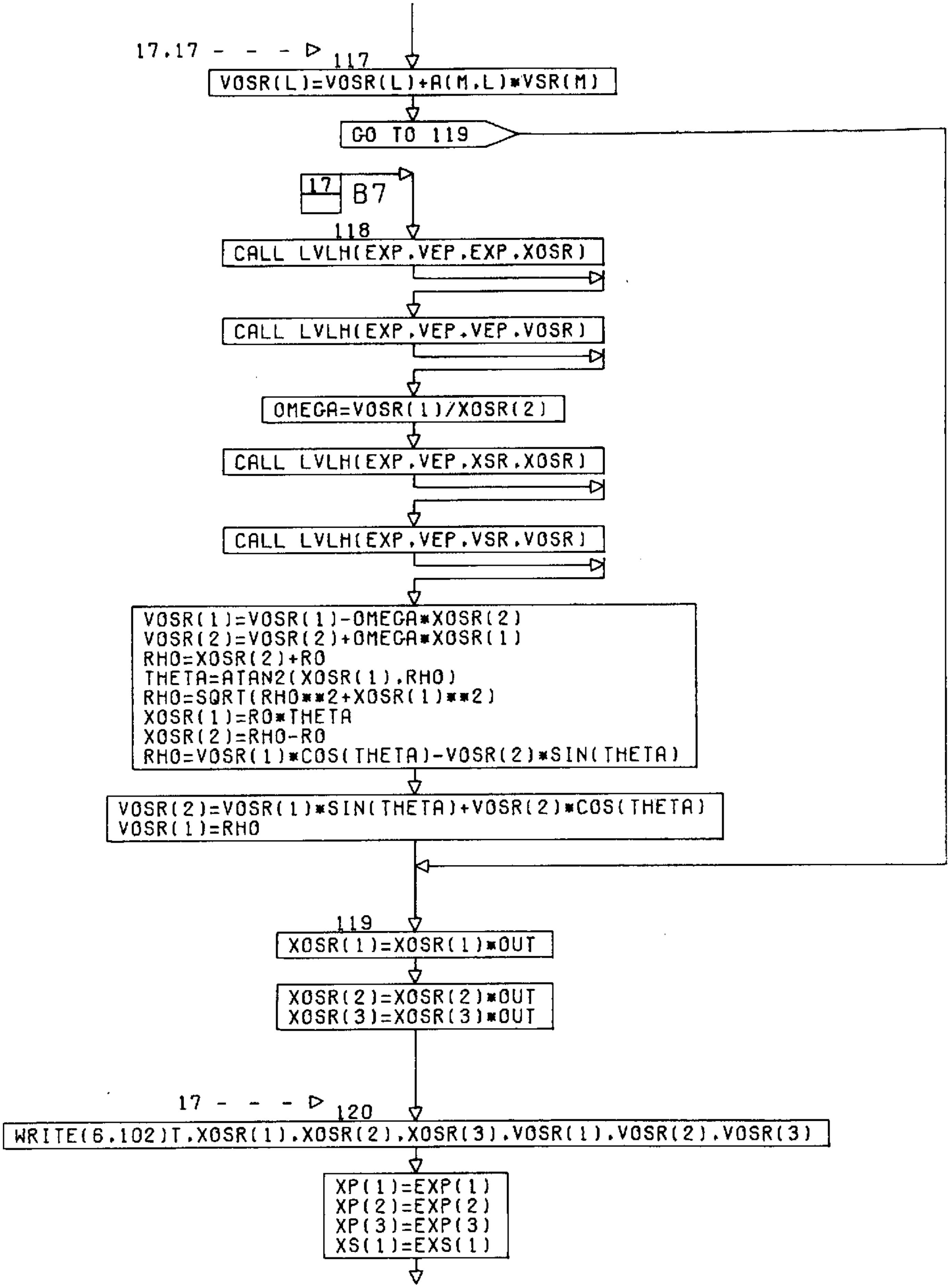
CONT. ON PG 16


15
B5
111
VSR(1)=VS(1)-VP(1)
VSR(2)=VS(2)-VP(2)
VSR(3)=VS(3)-VP(3)
IF(INOPT1-1)114,114,112
−
+
0
112
DO 113 J=1,3
VOSR(J)=0.0
DO 113 K=1,3
113
VOSR(J)=VOSR(J)+A(K,J)*VSR(K)
GO TO 115
17
B6
114
CALL LVLH(XP,VP,XP,VOSR)
OMEGA=VOSR(2)
CALL LVLH(XP,VP,VP,VOSR)
OMEGA=VOSR(1)/OMEGA
CALL LVLH(XP,VP,VSR,VOSR)
VOSR(1)=VOSR(1)-OMEGA*XOSR(2)/OUT
VOSR(2)=VOSR(2)+OMEGA*XOSR(1)/OUT
RHO=XOSR(2)+RO
THETA=ATAN2(XOSR(1),RHO)
RHO=SQRT(RHO**2+XOSR(1)**2)
XOSR(1)=RO*THETA
XOSR(2)=RHO-RO
RHO=VOSR(1)*COS(THETA)-VOSR(2)*SIN(THETA)

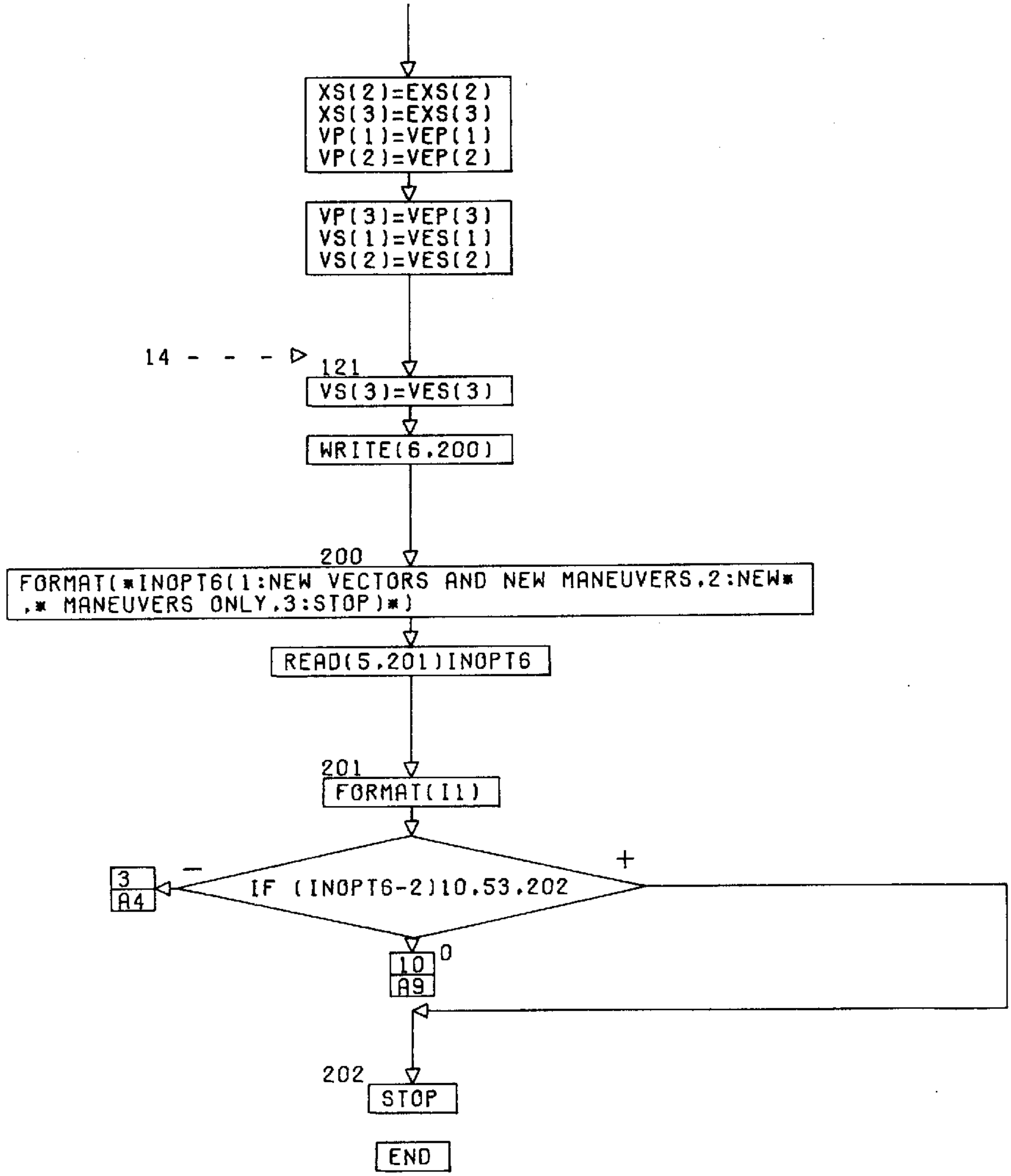
CONT. ON PG 17


VOSR(2)=VOSR(1)*SIN(THETA)+VOSR(2)*COS(THETA)
VOSR(1)=RHO
16
B6
115
WRITE(6,102)T,XOSR(1),XOSR(2),XOSR(3),VOSR(1),VOSR(2),VOSR(3)
DT=CT(I)/NP(I)
K=NP(I)
DO 120 J=1,K
18
T=T+DT
TPHASE=TPHASE+DT
CALL PROP(EMU,XP,VP,EXPE,TPHASE,EXP,VEP,XPO)
CALL PROP(EMU,XS,VS,EXES,TPHASE,EXS,VES,XSO)
EXPE=XPO
EXES=XSO
XSR(1)=EXS(1)-EXP(1)
XSR(2)=EXS(2)-EXP(2)
XSR(3)=EXS(3)-EXP(3)
VSR(1)=VES(1)-VEP(1)
VSR(2)=VES(2)-VEP(2)
VSR(3)=VES(3)-VEP(3)
−
IF(INOPT1-1)118,118,116
+
0
18
B7
116
DO 117 L=1,3
18
XOSR(L)=0.0
VOSR(L)=0.0
DO 117 M=1,3
18
XOSR(L)=XOSR(L)+A(M,L)*XSR(M)

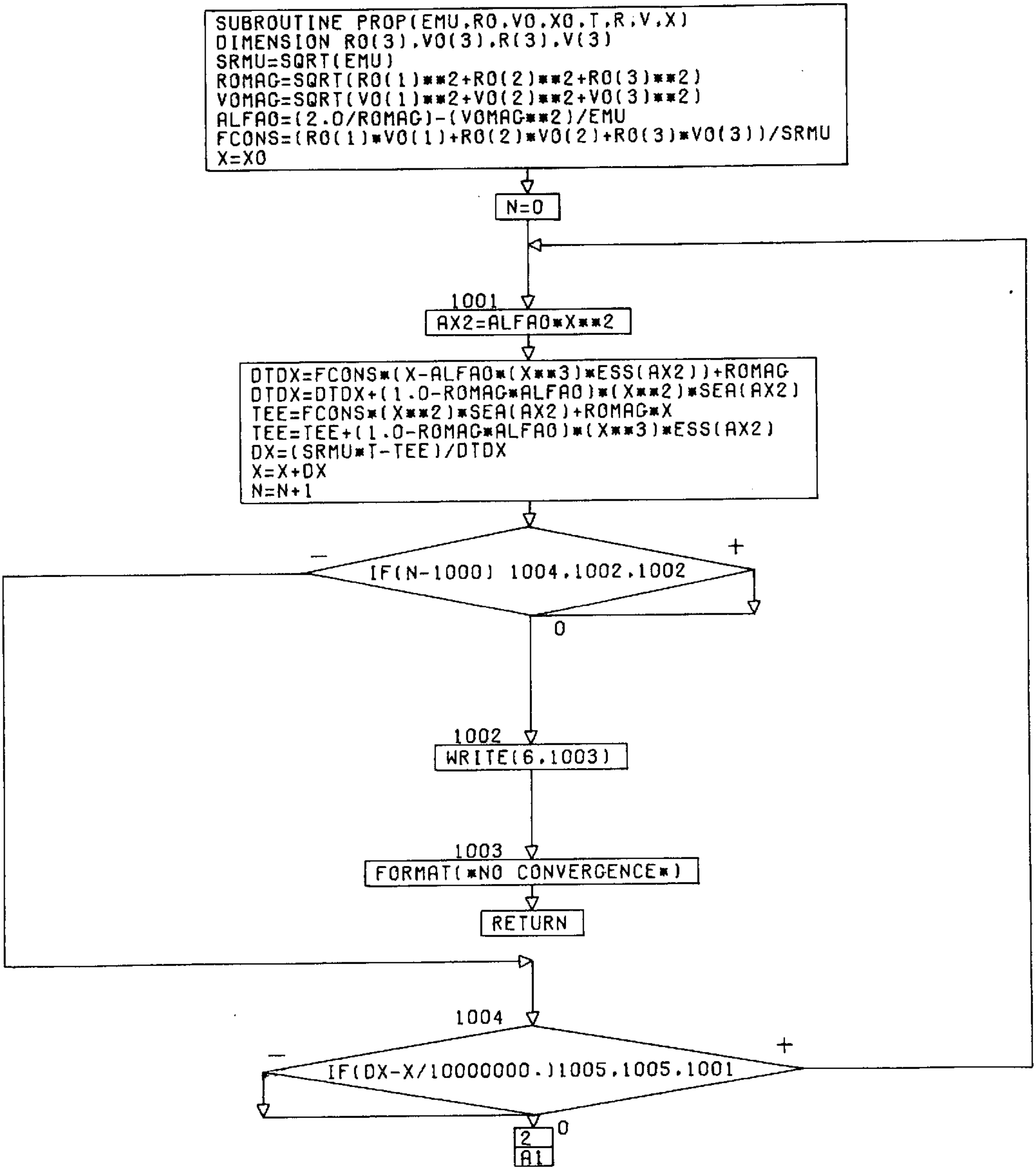
CONT. ON PG 18


17.17 - - - ▷
117
VOSR(L)=VOSR(L)+A(M,L)*VSR(M)
GO TO 119
17
B7
118
CALL LVLH(EXP,VEP,EXP,XOSR)
CALL LVLH(EXP,VEP,VEP,VOSR)
OMEGA=VOSR(1)/XOSR(2)
CALL LVLH(EXP,VEP,XSR,XOSR)
CALL LVLH(EXP,VEP,VSR,VOSR)
VOSR(1)=VOSR(1)-OMEGA*XOSR(2)
VOSR(2)=VOSR(2)+OMEGA*XOSR(1)
RHO=XOSR(2)+RO
THETA=ATAN2(XOSR(1),RHO)
RHO=SQRT(RHO**2+XOSR(1)**2)
XOSR(1)=RO*THETA
XOSR(2)=RHO-RO
RHO=VOSR(1)*COS(THETA)-VOSR(2)*SIN(THETA)
VOSR(2)=VOSR(1)*SIN(THETA)+VOSR(2)*COS(THETA)
VOSR(1)=RHO
119
XOSR(1)=XOSR(1)*OUT
XOSR(2)=XOSR(2)*OUT
XOSR(3)=XOSR(3)*OUT
17 - - - ▷
120
WRITE(6,102)T,XOSR(1),XOSR(2),XOSR(3),VOSR(1),VOSR(2),VOSR(3)
XP(1)=EXP(1)
XP(2)=EXP(2)
XP(3)=EXP(3)
XS(1)=EXS(1)

CONT. ON PG 19



XS(2)=EXS(2)
XS(3)=EXS(3)
VP(1)=VEP(1)
VP(2)=VEP(2)
VP(3)=VEP(3)
VS(1)=VES(1)
VS(2)=VES(2)
14
121
VS(3)=VES(3)
WRITE(6,200)
200
FORMAT(*INOPT6(1:NEW VECTORS AND NEW MANEUVERS,2:NEW*
,* MANEUVERS ONLY,3:STOP)*)
READ(5,201)INOPT6
201
FORMAT(I1)
IF (INOPT6-2)10,53,202
−
+
0
3
A4
10
A9
202
STOP
END

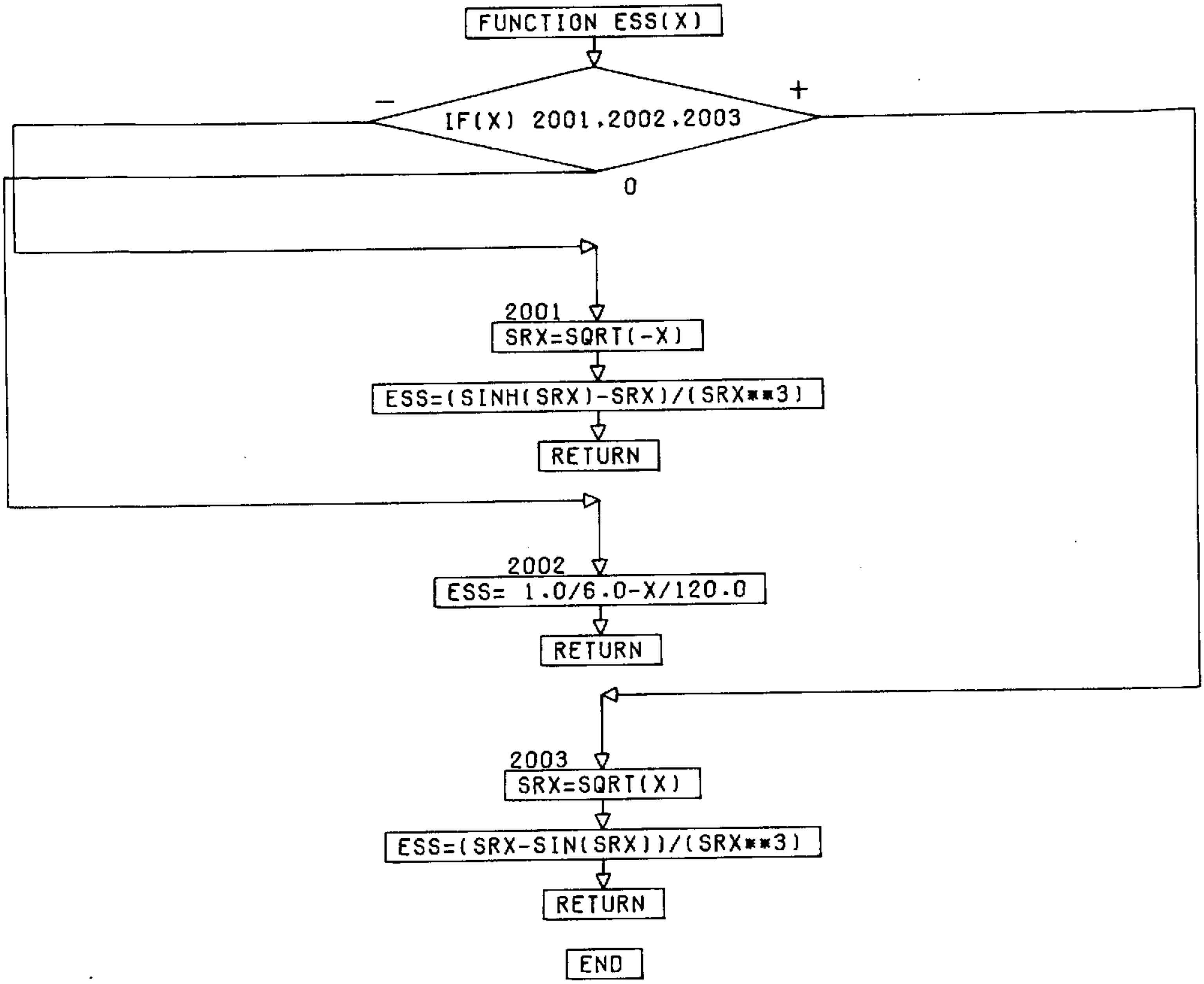

SUBROUTINE PROP(EMU,RO,VO,XO,T,R,V,X)
DIMENSION RO(3),VO(3),R(3),V(3)
SRMU=SQRT(EMU)
ROMAG=SQRT(RO(1)**2+RO(2)**2+RO(3)**2)
VOMAG=SQRT(VO(1)**2+VO(2)**2+VO(3)**2)
ALFAO=(2.0/ROMAG)-(VOMAG**2)/EMU
FCONS=(RO(1)*VO(1)+RO(2)*VO(2)+RO(3)*VO(3))/SRMU
X=XO
N=0
1001
AX2=ALFAO*X**2
DTDX=FCONS*(X-ALFAO*(X**3)*ESS(AX2))+ROMAG
DTDX=DTDX+(1.0-ROMAG*ALFAO)*(X**2)*SEA(AX2)
TEE=FCONS*(X**2)*SEA(AX2)+ROMAG*X
TEE=TEE+(1.0-ROMAG*ALFAO)*(X**3)*ESS(AX2)
DX=(SRMU*T-TEE)/DTDX
X=X+DX
N=N+1
IF(N-1000) 1004,1002,1002
−
+
0
1002
WRITE(6,1003)
1003
FORMAT(*NO CONVERGENCE*)
RETURN
1004
IF(DX-X/10000000.)1005,1005,1001
−
+
0
2
A1

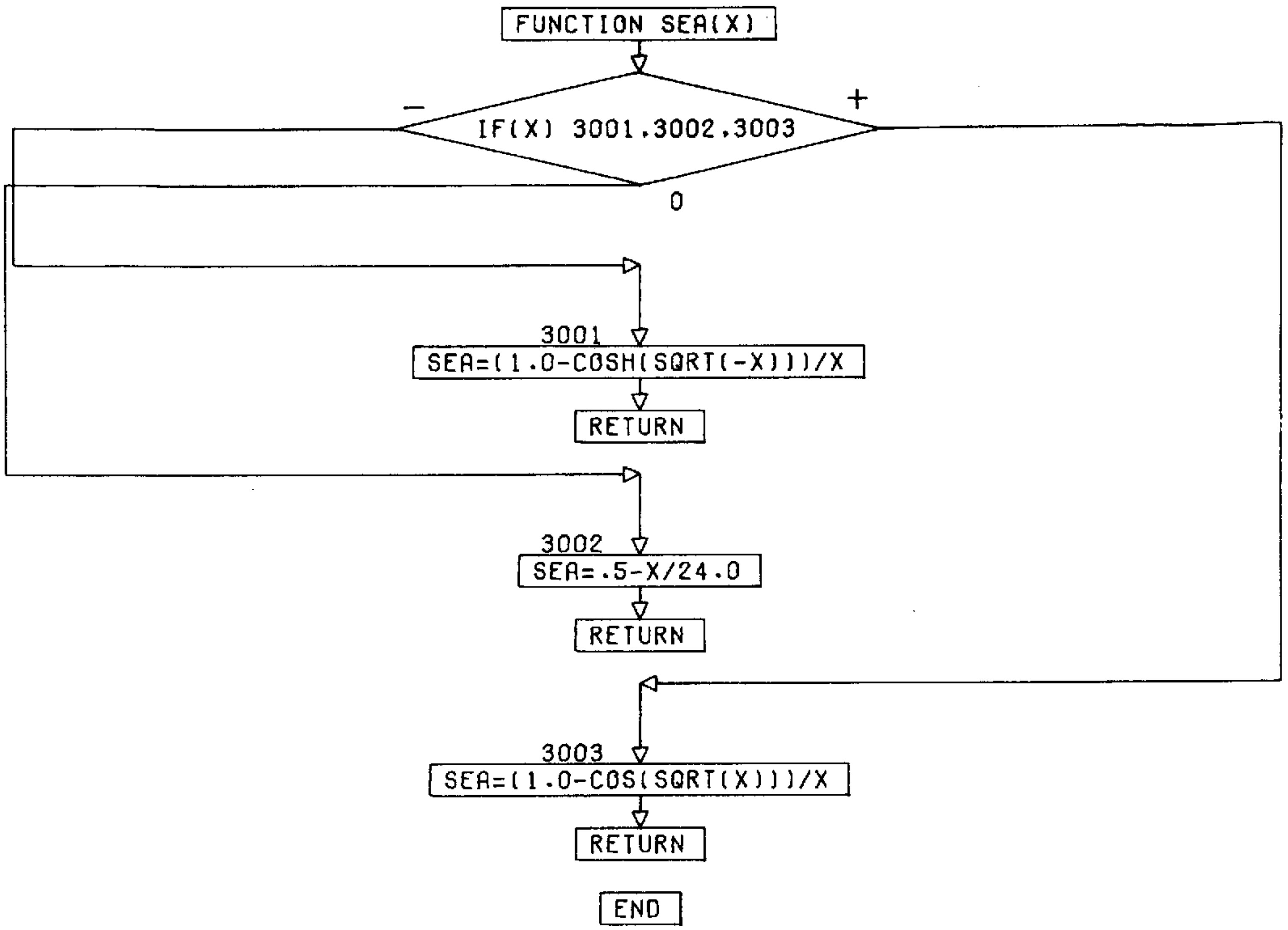
CONT. ON PG 2

```
A1
1005
AX2=ALFAO*X**2

A=1.0-(X**2)*SEA(AX2)/ROMAG
B=T-(X**3)*ESS(AX2)/SRMU
R(1)=A*RO(1)+B*VO(1)
R(2)=A*RO(2)+B*VO(2)
R(3)=A*RO(3)+B*VO(3)
RHO=SQRT(R(1)**2+R(2)**2+R(3)**2)
A=SRMU*(ALFAO*(X**3)*ESS(AX2)-X)/(RHO*ROMAG)
B=1.0-(X**2)*SEA(AX2)/RHO

V(1)=A*RO(1)+B*VO(1)
V(2)=A*RO(2)+B*VO(2)
V(3)=A*RO(3)+B*VO(3)

RETURN

END
```

```
FUNCTION SINH(X)
E=2.718281828459045
SINH=0.5*(E**X-E**(-X))

RETURN

END
```

```
FUNCTION COSH(X)
E=2.718281828459045
COSH=0.5*(E**X+E**(-X))

RETURN

END
```


FUNCTION ESS(X)
IF(X) 2001,2002,2003
−
+
0
2001
SRX=SQRT(-X)
ESS=(SINH(SRX)-SRX)/(SRX**3)
RETURN
2002
ESS= 1.0/6.0-X/120.0
RETURN
2003
SRX=SQRT(X)
ESS=(SRX-SIN(SRX))/(SRX**3)
RETURN
END


FUNCTION SEA(X)
IF(X) 3001,3002,3003
−
+
0
3001
SEA=(1.0-COSH(SQRT(-X)))/X
RETURN
3002
SEA=.5-X/24.0
RETURN
3003
SEA=(1.0-COS(SQRT(X)))/X
RETURN
END

```
SUBROUTINE LVLH(X,V,A,B)
DIMENSION X(3),V(3),A(3),B(3)
AK1=X(3)*V(2)-X(2)*V(3)
AK2=X(1)*V(3)-X(3)*V(1)
AK3=X(2)*V(1)-X(1)*V(2)
AKM=SQRT(AK1**2+AK2**2+AK3**2)
AK1=AK1/AKM
AK2=AK2/AKM
```

```
AK3=AK3/AKM
R=SQRT(X(1)**2+X(2)**2+X(3)**2)
AJ1=X(1)/R
AJ2=X(2)/R
AJ3=X(3)/R
AI1=AJ2*AK3-AJ3*AK2
AI2=AJ3*AK1-AJ1*AK3
AI3=AJ1*AK2-AJ2*AK1
```

```
B(1)=A(1)*AI1+A(2)*AI2+A(3)*AI3
B(2)=A(1)*AJ1+A(2)*AJ2+A(3)*AJ3
B(3)=A(1)*AK1+A(2)*AK2+A(3)*AK3
```

```
RETURN
```

```
END
```

```
SUBROUTINE ECI(X,V,A,B)
DIMENSION X(3),V(3),A(3),B(3)
AK1=V(2)*X(3)-V(3)*X(2)
AK2=V(3)*X(1)-V(1)*X(3)
AK3=V(1)*X(2)-V(2)*X(1)
AKM=SQRT(AK1**2+AK2**2+AK3**2)
AK1=AK1/AKM
AK2=AK2/AKM
```

```
AK3=AK3/AKM
R=SQRT(X(1)**2+X(2)**2+X(3)**2)
AJ1=X(1)/R
AJ2=X(2)/R
AJ3=X(3)/R
AI1=AJ2*AK3-AJ3*AK2
AI2=AJ3*AK1-AJ1*AK3
AI3=AJ1*AK2-AJ2*AK1
```

```
B(1)=A(1)*AI1+A(2)*AJ1+A(3)*AK1
B(2)=A(1)*AI2+A(2)*AJ2+A(3)*AK2
B(3)=A(1)*AI3+A(2)*AJ3+A(3)*AK3
```

```
RETURN
```

```
END
```